UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER TD-CR-77-2	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) MCARLO: A COMPUTER PROGRAM FOR GENERATING MONTE-CARLO TRAJECTORIES IN A TIME-VARYING MAN/MACHINE CONTROL TASK	5. TYPE OF REPORT & PERIOD COVERED Contractor Report
	6. PERFORMING ORG. REPORT NUMBER 3463
7. AUTHOR(s) David L. Kleinman, Sheldon Baron, and Jeffrey E. Berliner	8. CONTRACT OR GRANT NUMBER(s) DAAH01-76-C-0194
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, Massachusetts 02138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, US Army Missile R&D Command ATTN: DRDMI-TI Redstone Arsenal, AL 35809	12. REPORT DATE 10 Jun 1977
	13. NUMBER OF PAGES 105
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Commander, US Army Missile R&D Command ATTN: DRDMI-TD Redstone Arsenal, AL 35809	15. SECURITY CLASS. (of this report) UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

MCARLO
Fortran
TIVAR

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

MCARLO is a computer program for generating time histories of pertinent variables in a man-machine control task. The optimal control model forms the basis for the Monte-Carlo simulation equations. This report gives the modeling formulation and requisite discretization of the equations, a description of the MCARLO subroutines, and input deck setup and a sample problem with solution.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

U.S. ARMY MISSILE RESEARCH AND DEVELOPMENT COMMAND

TECHNICAL REPORT TD-CR-77-2

MCARLO: A COMPUTER PROGRAM FOR GENERATING MONTE-CARLO TRAJECTORIES IN A TIME-VARYING MAN/MACHINE CONTROL TASK

Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02138

10 June 1977

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Prepared for:
Aeroballistics Directorate
Technology Laboratory

Redstone Arsenal, Alabama 35809

DMI FORM 1000, 1 APR 77

DISPOSITION INSTRUCTIONS

DISCLAIMER

TRADE NAMES

TABLE OF CONTENTS

1. COMPUTER PROGRAM ABSTRACT

PROGRAM NAME: MCARLO

ORIGINATOR: Bolt Beranek and Newman Inc.
50 Moulton Street
Cambridge, Massachusetts 02138
(617) 491-1850
David L. Kleinman, Sheldon Baron,
and Jeffrey E. Berliner

CONTRACT MONITOR: U. S. Army Missile Command
Aeroballistics Directorate
(205) 876-1951
Richard E. Dickson

CONTRACT NUMBER: DAAH01-76-C-0194

PROGRAM ABSTRACT

MCARLO is a computer program for generating simulated time histories of pertinent variables in a man-machine control task. The optimal control model (OCM) forms the basis for the Monte-Carlo simulation equations. The ensemble statistics of such time functions must agree with the covariance propagation results obtained via more direct methods, e.g., using TIVAR.

The MCARLO program is written in the FORTRAN-IV-EXTENDED computer programming language, and is designed for efficient batch operation on a Control Data CDC-6600 computer. Data input to the program is provided on standard punched cards and output is generated via the lineprinter.

In this manual we give the modeling formulation and requisite discretization of the equations, a description of the MCARLO subroutines, the input deck setup and a sample problem with solution.

2. PROBLEM FORMULATION AND ALGORITHMS

The major aspects of computer simulation for a man-machine system are shown in Figure 1. The modeling issues are discussed below.

2.1 System-Display Dynamics

There are no modeling restrictions on the system being controlled, other than the generation of a set of NY displayed elements y_t at time t from:

$u(t)$ = human's control inputs, NU vector

$w(t)$ = random input disturbances, NW vector

$z(t)$ = "deterministic" inputs, NZ vector.

In its most general mathematical form, the system/display dynamics might be modeled by:

$$\dot{x}(t) = f(t, x(t), u(t), w(t), z(t)); \quad x_o = x(t_o) \tag{1}$$

$$y(t) = h(t, x(t), u(t)) \tag{2}$$

where $x(t)$= NX system state vector. The vector $w(t)$ consists of independent zero mean white Gaussian noise inputs with covariance

$$E[w_i(t)\ w_i(\sigma)] = W^o_{si}(t)\ \delta(t-\sigma) \quad i=1,\ldots,NW \tag{3}$$

For the special case of a linear time-varying system, the equations (1)-(2) become:

$$\dot{x}(t) = A_s x(t) + B_s u(t) + E_s w(t) + F_s z(t) \tag{4}$$

$$y(t) = C_s x(t) + D_s u(t) \tag{5}$$

The system parameters, which may be time varying, are:

A_s = NX by NX state matrix

B_s = NX by NU control matrix

E_s = NX by NW noise matrix

F_s = NX by NZ bias input matrix

C_s = NY by NX state display matrix

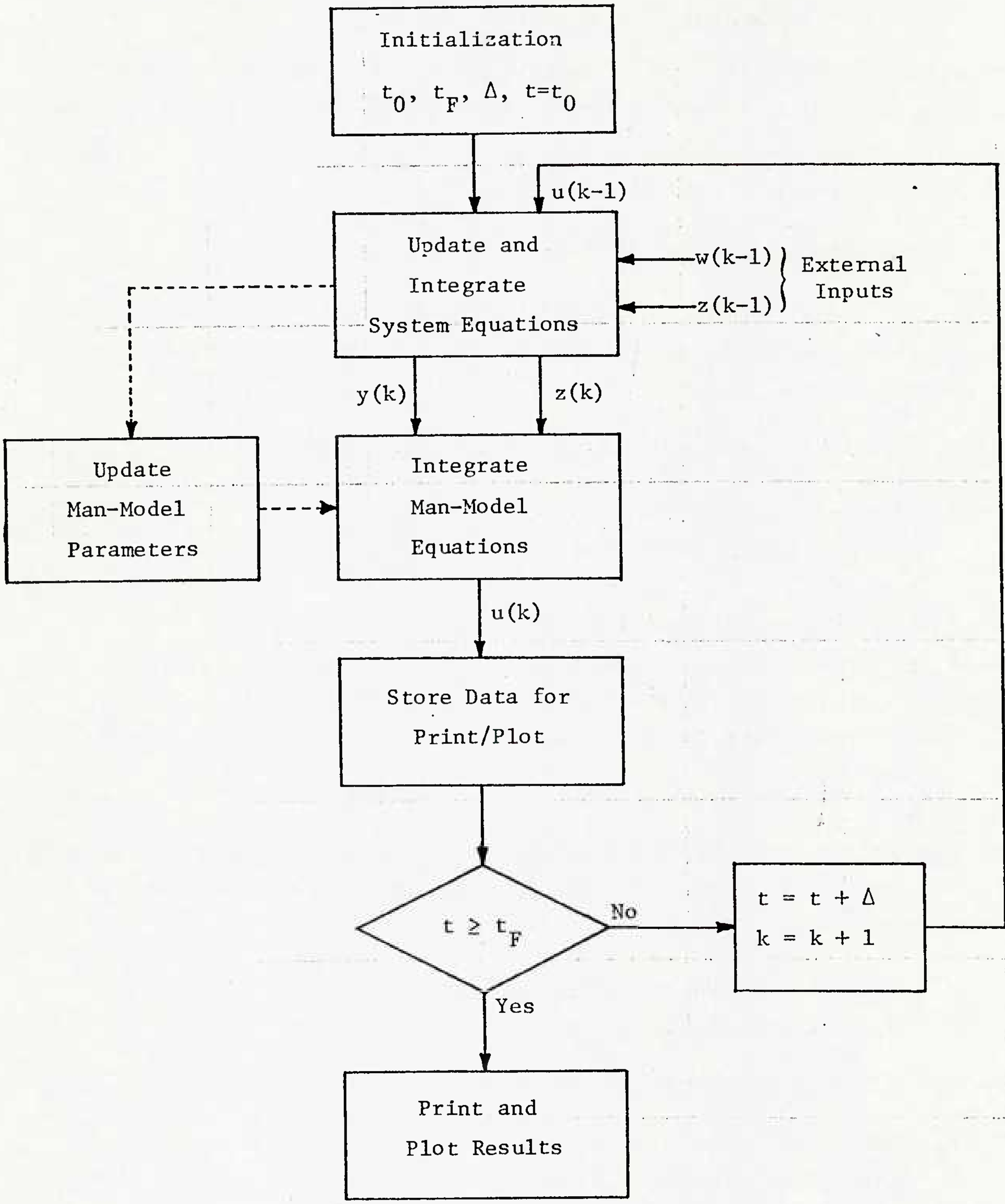


Figure 1. Major Aspects of Computer Simulation for a Man-Machine System

JEB-7

D_s = NY by NU control display matrix

A means for updating the system parameters, as a function of time, must be included along with the system description. Since the form of updating is highly dependent upon the system model, a general updating scheme is feasible for only the highly structured linear case.

2.2 Human Operator Internal Model

In the OCM, the human is assumed to have an internal characterization of the input-output response of the system. This "internal model" is assumed to be linear, in state variable form,

$$\dot{x}_m(t) = A_m x_m(t) + B_m u_m(t) + E_m w_m(t) + F_m z_m(t) \tag{6}$$

$$y_m(t) = C_m x_m(t) + D_m u_m(t) \tag{7}$$

where

$x_m(t)$ = internal model states, NXM vector
$z_m(t)$ = model deterministic inputs, NZM vector
(present formulation requires z_m=z so NZM=NZ)
$w_m(t)$ = model Gaussian white noise inputs, NWM vector

$$E[w_{mi}(t)\ w_{mi}(\sigma)] = W^o_{mi}(t)\ \delta(t-\sigma); \quad i=1,\ldots,NWM \tag{8}$$

The model inputs $u_m(t)$ and displayed outputs $y_m(t)$ are assumed to be the same as the actual system inputs u(t) and displays y(t) to avoid numerous conceptual problems. Thus,

$u_m(t) = u(t)$ and NUM = NU
$y_m(t) = y(t)$ and NYM = NY.

The internal model parameters, which can be time-varying, are:

A_m = NXM by NXM model state matrix
B_m = NXM by NUM model control matrix
E_m = NXM by NWM model noise matrix
F_m = NXM by NZM model bias input matrix

C_m = NYM by NXM model output matrix for states

D_m = NYM by NUM model output matrix for controls

The choice of model parameter matrices is somewhat subjective. In the general non-linear case, these matrices typically would approximate the partial derivatives of f and h in Equations (1)-(2), i.e.,

$$A_m \simeq \frac{\partial f}{\partial x} \text{ , etc.}$$

In the linear system case of Equations (4)-(5), the model typically would reflect an appropriate lower order characterization of the true system dynamics. Of course, a not unreasonable choice for the model matrices is:

$$A_m = A_s \text{ , etc.}$$

Here, the model parameter is assumed to be same as the associated system parameter. This is a convenient assumption as it greatly simplifies the process of updating model matrices for time varying systems.

The internal model is used within the OCM to help generate a (continuous time) human operator control input via:

$$\dot{u}(t) = -L_c \begin{bmatrix} \hat{x}_m(t) \\ u(t) \end{bmatrix} + L_{c2}\, v_u(t) \tag{9}$$

The NUM by (NXM+NUM) feedback gains

$$L_c = [T_N^{-1} L_{opt} \mid T_N^{-1}] = [L_{c1} \mid L_{c2}] \tag{10}$$

are generated via auxiliary programs that solve the optimal control problem for the model equations. The model is also needed in the construction of the Kalman filter-predictor that generates the model state estimate $\hat{x}_m(t)$.

2.3 Human Limitations

The human generates $\hat{x}_m(t)$ on the basis of the delayed and noisy perceived information:

$$y_{pi}(t) = N_i[y_i(t-\tau)] + v_{yi}(t) \qquad i=1,\ldots,NY \tag{11}$$

where

τ = the human's time delay,

$v_y(t)$ = the observation or sensor noise at time t,

and $N_i(.)$ is the non-linear observation threshold:

$$N_i(x) = \begin{cases} x-a_i & x > a_i \\ 0 & |x| \leq a_i \\ x+a_i & x < -a_i \end{cases} \tag{12}$$

In a simulation model, it is possible to implement the non-linear observations using Equations (11) and (12). However, in a man-machine context we find it more convenient to replace $N_i(.)$ by an equivalent gain, $\hat{N}_i$. The random input describing function:

$$\hat{N}_i = \operatorname{erfc} \frac{|x|}{a_i\sqrt{2}} \tag{13}$$

is used. N_i is interpreted as the <u>probability</u> that the human will respond to y_i, given its present value at time t.

Each sensor noise $v_{yi}(t)$ is a zero-mean, white Gaussian noise with covariance:

$$E[v_{yi}(t)\ vyi(\sigma)] = \frac{V^o_{yi}(t)}{f_{i(t)}}\delta(t-\sigma) \tag{14}$$

that contains both an <u>additive</u> and a <u>ratioed</u> component:

$$V^o_{yi}(t) = V_{yi}(t) + \pi\rho_{yi}E[y_i 2(t- \ )] \tag{15}$$

The quantity $f_i > 0$ is the attentional allocation to the displayed variable yi. The f_i are constrained by:

$$\frac{1}{2}\sum_{i=1}^{NY} f_i(t) = f_T = \text{constant total attention} \tag{16a}$$

$$f_{i+1}(t) = f_i(t) \quad i=1,3,\ldots,NY-1 \tag{16b}$$

to indicate that position-velocity pairs are obtained simultaneously form the display elements.

The neuro-motor interface portion of the model is given by Equation (9). The motor noises $v_{ui}(t)$, $i=1,\ldots,NU$ are zero-mean white Gaussian, with covariance:

$$E[v_{ui}(t)\ v_{ui}(\sigma)] = V^{o}_{ui}(t)\ \delta(t-\sigma) \tag{17}$$

that contains an additive and a ratioed component,

$$V^{o}_{ui}(t) = V_{ui}(t) + \pi\rho_{ui}\ \mathrm{Var}[u_i(t)] \tag{18}$$

2.4 Discretized Equations

The implementation of the human operator simulation on a digital computer requires the discretization of both system and model equations. Given a computer time step Δ, the system must generate $y(k) \equiv y(t_o+k\Delta) = y(t)$ from the inputs $u(k-1)$, $w(k-1)$, and $z(k-1)$ which are assumed to be piecewise-constant over the previous time interval, e.g.,

$$u(t) = u(k-1) \qquad t_o+(k-1)\Delta < t \leq t_o+k\Delta \tag{19}$$

For the case in which the system is described by the linear equations (4)-(5), the discretization chosen is:

$$x(k+1) = \Phi_s x(k) + \Gamma_s u(k) + \Delta[E_s w(k) + F_s z(k)] \tag{20}$$

$$y(k) = C_s x(k) + D_s u(k-1) \tag{21}$$

where $x(0) = x(t_o)$ = initial state. The discrete system matrices Φ_s and Γ_s are:

$$\Phi_s = e^{A_s\Delta} \quad ; \quad \Gamma_s = \int_0^{\Delta} e^{A_s\sigma} B_s d\sigma \tag{22}$$

Discretization of the linear model equations (6)-(7) is done in a manner similar to the above. Thus

$$x_m(k+1) = \Phi_m x_m(k) + \Gamma_m u(k) + \Delta[E_m w_m + F_m z_m(k)] \tag{23}$$

$$y(k) = C_m x_m(k) + D_m u(k-1) \tag{24}$$

where

$$\Phi_m = e^{A_m \Delta} \quad ; \quad \Gamma_m = \int_0^{\Delta} e^{A_m \sigma} B_m d\sigma \tag{25}$$

The human operator model must generate a control input u(k), to use over the time interval $(t, t+\Delta)$, via

$$\frac{u(k) - u(k-1)}{\Delta} = -L \begin{bmatrix} \hat{x}_m(k) \\ u(k-1) \end{bmatrix} + L_2 v_u(k) \tag{26}$$

Note that it is the control input itself that is considered to be piecewise constant for interface with the system model. This is in contrast to the covariance propagation approach where control-rate is assumed piecewise constant with:

$$\dot{u}(k) = -L_d \begin{bmatrix} \hat{x}(k) \\ u(k) \end{bmatrix} + L_{d2} v_u(k) \tag{27}$$

The gains $L=[L_1 \vdots L_2]$ in Equation (26) are computed from either the gains L_d or L_c according to

$$L = \frac{1}{2} L_d \tag{28a}$$

or

$$L = \frac{1}{2} \hat{L}_d \tag{28b}$$

respectively, where $\hat{L}_d$ are the equivalent discrete gains:

$$\hat{L}_d = L_c \left[\frac{1}{\delta} \int_0^{\delta} e^{\bar{A} \sigma} d\sigma \right] \tag{29}$$

$$\bar{A} = \left[\begin{array}{c|c} A_m & B_m \\ \hline -L_{c1} & -L_{c2} \end{array} \right]$$

The discretized observations are:

$$y_{pi}(k) = \hat{N}_i \, y_i(k-N) + v_{yi}(k) \tag{30}$$

where N= integer[τ/Δ] and the covariance of the piecewise-constant white noise $v_{yi}(k)$ is $\Delta^{-1}[V^o_{yi}(k)/f_i(k)]$ to account for the finite time step. Similarly, the covariance of the motor noise $v_u(k)$ now becomes V^o_{ui}/Δ.

2.5 Human Operator Model Equations

Equations (23), (24), and (26) may be combined into an augmented man-model equation, suitable for Monte-Carlo simulation. Defining the augmented state $x(k) = [x(k),u(k-1)]$, and input $w = [w_m,v_u]$, we obtain

$$\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma u_c(k) + Ew(k) + Fz_m(k) \tag{31}$$

$$y(k) = C\underline{x}(k) \tag{32}$$

where $u_c(k) = L_1\hat{x}_m(k)$ is the "commanded" control. The augmented matrices are obtained by rewriting Equation (26),

$$u(k) = (I-\Delta L_2)u(k-1) + \Delta uc(k) + \Delta L2vu(k) \tag{26a}$$

and combining with Equation (23), yielding:

$$\Gamma = \Delta \begin{bmatrix} \Gamma_m \\ \hline I \end{bmatrix} \; ; \quad F = \Delta \begin{bmatrix} F_m \\ \hline 0 \end{bmatrix} \; ; \quad E = \Delta \left[\begin{array}{c|c} E_m & \Gamma_m L_2 \\ \hline 0 & L_2 \end{array}\right]$$

$$\Phi = \left[\begin{array}{c|c} \Phi_m & \Gamma_m(I-\Delta L_2) \\ \hline 0 & (I-\Delta L_2) \end{array}\right] \; ; \quad C = \left[\begin{array}{c|c} C_m & D_m \end{array}\right] \tag{33}$$

The equations (31)-(32) are similar to the discretized equations that occur in the covariance propagation studies using the OCM. Borrowing heavily from earlier efforts, it is easy to write the equations for the Kalman filter-predictor combination that generates the state estimate $\hat{x}(k)$. For compatibility with the covariance propagation modeling, we use the a posteriori estimate

$$\hat{\underline{x}}(k) = \hat{\underline{x}}(k|k) = E[\underline{x}(k)|y_p(0), \ldots, y_p(k)] \tag{34}$$

The Kalman filter generates the (a posteriori) estimate of the delayed state,

$$\hat{p}(k|k) = E[\underline{x}(k-N)|y_p(0), \dots ,y_p(k)] \tag{35a}$$

and the (a priori) one-step ahead prediction

$$\hat{p}(k+1|k) = E[\underline{x}(k+1-N)|y_p(0), \dots ,y_p(k)] \tag{35b}$$

by means of the usual update and propagate set of equations. These are, respectively:

$$\hat{p}(k|k) = \hat{p}(k|k-1) + G_k\, \nu(k) \tag{36a}$$

$$\hat{p}(k+1|k) = \Phi\, \hat{p}(k|k) + \Gamma u_c(k-N) \tag{36b}$$

where $\nu(k)$ is the innovations, or residual sequence

$$\nu(k) = y_p(k) - C\, \hat{p}(k|k-1) \tag{37}$$

and the initial condition is

$$\hat{p}(0|-1) = \text{given} = \underline{x}(0) \tag{38}$$

The filter gain G_k is[1]

$$G_k = {}_{k|k-1}C'[C\, {}_{k|k-1}C' + \frac{1}{\Delta}V_y(k)]^{-1} \tag{39}$$

where the update-propagate sequence for generating the Riccati solution Σ is:

$$\Sigma_{k|k} = (I-G_kC)\, \Sigma_{k|k-1}\, (I-G_kC)' + G_k\frac{\bar{V}_y(k)}{\Delta}G_k' \tag{40a}$$

$$\Sigma_{k|k} = \Phi\Sigma_{k|k}\Phi' + EWE' + FZF' \tag{40b}$$

and

$$W = \text{diag}[w_{mi}(k)/\Delta, Vo_{ui}(k)/\Delta]$$

$$Z = \text{diag}[\frac{T_{cor}}{\Delta}\, z_i^2(k)]$$

are "pseudo-noise" covariance matrices. The "correlation time", T_{cor} = 1 sec.

The predictor forms the estimate $\hat{\underline{x}}(k)$ from $\hat{p}(k|k)$ using:

---

(1) For simplicity, the RIDF gain N_i is included with v_y, rather than with C. This is the usual practice in the OCM.

$$\underline{\hat{x}}(k) = \Phi^N \hat{p}(k|k) + \sum_{i=0}^{N-1} \Phi^i \Gamma u_c(k-i-1) \qquad N>0 \tag{41}$$

If N=0, $\underline{\hat{x}}(k) = \hat{p}(k|k)$.

2.6 Special Considerations

There are several issues in the simulation of the above equations that remain to be resolved. Some are unique to the man-machine problem.

2.6.1 Storage of Delayed Quantities

The simulation equation (41) requires knowledge of the commanded control $u_c(k-1),\ldots,u_c(k-N)$. The update equation (36b) requires $u_c(k-N)$. Similarly, the computation of $\nu(k)$ is based on the delayed quantity $y(k-N)$. To meet these requirements we retain in storage, at time k,

$$\text{PASTUC} = [u_c(k-N) \;\ldots\; u_c(k)] \tag{42a}$$

$$\text{PASTY} = [y(k-N) \;\ldots\; y(k)] \tag{42b}$$

2.6.2 On-line Variance Estimation

The diagonal observation noise covariance matrix $\bar{V}_y$ in Equations (39) and (40a) is given by:

$$\bar{V}_{yi}(k) = \frac{V_y^o(k)}{f_i(k)\ \hat{N}_i^2(k-N)} \qquad i=1,\ldots,\text{NYM} \tag{43}$$

where $f_i(k)$ is the fractional attention to y_{pi} at time $k \geq N$, and:

$$V_{yi}^o(k) = V_{yi}(k) + \pi \rho_{yi} E[y_i^2(k-N)] \tag{44}$$

$$\hat{N}_i(k-N) = \operatorname{erfc}\left[\frac{|y_i(k-n)|}{a_i\sqrt{2}}\right]$$

Similarly, the covariance of the motor noise:

$$V_{ui}^o(k) = V_{ui}(k) + \pi \rho_{ui} \operatorname{Var}[u_i(k-1)] \qquad i=1,\ldots,\text{NUM} \tag{45}$$

Both Equations (44) and (45) require process (i.e. ensemble) <u>statistics</u> at time k. However, these are not available from a single Monte-Carlo trajectory, and their precomputation for subsequent read-in is unfeasible. The approach we have taken is to obtain temporal approximations using filtered past data. An approximation

$$\alpha(k) \doteq E[y_i^2(k-N)]$$

is obtained via 1st-order filtering of $y_i^2(k-N)$,

$$\alpha(k) = e^{-\Delta/\tau_m}\alpha(k-1) + (1 - e^{-\Delta/\tau_m})y_i^2(k-N) \tag{46}$$

with initial condition $\alpha(N-1) = y_i^2(0)$. The approximate variance of $u_i^2(k-1)$ is found using a two-step procedure that estimates (through filtering) the mean and mean-square, and then computes the variance. The time constant $\tau_m = 0.5$ sec.

2.6.3 Pseudo-Random Noise Sequence

The Monte-Carlo simulation of the human operator equations must generate discrete white-noise sequences for observation noise and motor noise that have specified variances $\bar{V}_{yi}/\Delta$ and V^o_{ui}/Δ , respectively. This accomplished by picking, at time k,

$$V_{ui}(k) = \frac{V^o_{ui}}{\Delta}^{1/2} \xi(k)$$

where $\xi(k)=N(0,1)$ is a unit variance, zero mean, Gaussian random variable. (k) is generated by averaging 12 uniformly distributed (-1/2, +1/2) independent random variables. A slight modification is made when choosing the observation noise sequence, to reflect more precisely the underlying multiplicative noise process. We pick

$$v_{yi}(k) = \frac{\tilde{V}_{yi}}{\Delta}^{1/2} \xi(k)$$

where

$$\tilde{V}_{yi}(k) = \frac{V_{yi} + \pi \rho_i y_i^2(k-n)}{f_i(k)\ \hat{N}_i^2(k-N)}$$

depends only on the <u>instantaneous</u> value of $y_i(k-N)$.

2.7 Summary of Human Model Computations

The major part of the man-machine simulation is the implementation of the equations of the OCM, where a control input u_k is generated from the observations y_k. The steps in this process are summarized below.

1. Storage of $y(k)$ in the (N+1)-st column of PASTY, Eq. (42b).

2. Computation of $\alpha(k)$ via Eq. (46), and the observation noise covariance $\bar{V}_y(k)$ using Eqs. (43)-(44).

3. Computation of the residuals

$$\nu(k) = y(k-N) + v_y(k) - C\hat{p}(k|k-1) \tag{47}$$

4. Compute the filter gain G_k via Eq. (3a) and update the Riccati equation (40a) to obtain $\Sigma_{k|k}$.

5. Obtain the a posteriori estimate $\hat{p}(k|k)$, Eq. (36a).

6. Obtain the state estimate $\hat{\underline{x}}(k)$ via Eq. (41).

7. Compute the new commanded control,

   $$u_c(k) = -L_1\hat{x}(k)$$

   and store it in the (N+1)-st column of PASTUC.

8. Generate the piecewise constant control $u(k)$ to use over the upcoming time interval $[k,k+1]$ from Eq. (26a).

9. Update the estimate of $\mathrm{Var}[u(k)]$ for use at the next time step.

10. Propagate Σ and $\hat{p}$ using Eqs. (40b) and (36b).

11. Do a stack pushdown (i.e., column shift to the left) on PASTY and PASTUC to get ready for the next time step.

3. PROGRAM DESCRIPTION

The computer program MCARLO has been developed for simulating the human operator equations, and controlling the parameter updating processes. The program consists of six major routines that are highly modular in structure:

1. MAIN
2. SYSTM
3. UPDATE
4. MAN
5. INFORM
6. PRINTR

along with numerous minor, user supplied routines. The function of each of the above routines is discussed.

3.1 MAIN Program

The MAIN program initializes time and controls the overall program flow according to Figure 1. It calls the required subroutines for system propagation, and information storage. Time is incremented, $t = t+\Delta$, and the cycle repeats. When $t \geq t_f$, the printout routine is called.

3.2 Subroutine SYSTM

Subroutine SYSTM is a user-oriented routine that simulates the response of the actual system. Given the control input u, and the external or disturbance inputs w and z over the time interval $(t-\Delta, t]$, SYSTM returns the value of y at time t. At the first time step, $t = t_o$, only internal initializations are performed. The present implementation requires SYSTM to compute and return (at time t) the values of w and z for its own use over the next time interval. As an alternate approach, a separate subroutine EXTINP might perform this function once NW and NZ are known.

The SYSTM subroutine is entirely self-contained. As such it is possible to replace it, in its entirety, by user written routine that simulates the

given system.(2) The only requirement is the generation of y(t) from control input u(t) and the disturbance inputs. If the system is time-varying, the logic for updating system parameters must be included within the SYSTM routine. The system simulation can thus be as complicated or as simple as the problem may warrant.

The SYSTM subroutine now contained in the MCARLO program treats the general linear case

$$\dot{x}(t) = A_s x(t) + B_s u(t) + E_s w(t) + F_s z(t)$$
$$x(t^+) = x(t^-) + \delta x(t) \;; \quad x(t_o^-) = 0$$
$$y(t) = C_s x(t) + D_s u(t)$$

$$\mathrm{cov}[W_i(t)] = W^o_{si}(t)$$

where any parameter matrix can be time-varying. The method by which parameters are updated is similar to the alphanumeric code/index scheme used in TIVAR. Parameters can be changed at time t via external or card inputs. They can also be changed periodically via an internal -- user supplied -- subroutine SYSNEW. Table Ia defines the parameter codes for SYSTM. Note that mnemonics A and not AS, etc., have been used for compatibility with TIVAR deck setups.

At time t the subroutine updates the pertinent variables and (if necessary) computes new discretize equations. The state is propagated using the transition matrix method. At time t_o matrices A, B, C must be input for proper initialization. Unless input, D, E, F, W^o_s, x are assumed to be zero. Finally, the parameters associated with <u>codes 1-7</u> are stored in common blocks. This makes them accessible to other subroutines for the special case when <u>model = system</u>.

---

(2) An analog simulation or a "real" system with A/D interface provides an interesting possibility.

Table Ia: PARAMETER CODES IN SYSTM

CODE	KEY	DESCRIPTION
A	1	System A_s matrix, NX by NX
B	2	Control B_s matrix, NX by NU
C	3	Output C_s matrix, NY by NX
D	4	Output D_s matrix, NY by NU
E	5	Noise E_s matrix, NX by NW
F	6	Bias input F_s matrix NX by NZ
WO	7	Noise covariances Wo_{si}, NW vector
XINC	8	Increment δx to system state, NX vector
INT	9	Transfer to subroutine SYSNEW
PRINT	10	Printout interval for data

Table Ib: PARAMETER CODES IN UPDATE

CODE	KEY	DESCRIPTION
AM	1	Model A_m matrix, NXM by NXM
BM	2	Control B_m matrix, NXM by NUM
CM	3	Output C_m matrix, NYM by NXM
DM	4	Output D_m matrix, NYM by NUM
EM	5	Noise E_m matrix, NXM by NWM
FM	6	Bias input F_m matrix, NXM by NZM
WOM	7	Model noise covariances Wo_m, NWM vector
XHINC	8	Increment $\delta\hat{x}$ to $\hat{p}_{k\|k-1}$, NXM vector
TD	9	Human´s time delay τ
MNA	10	Additive NUM motor noise variances V_u
MNR	11	Motor noise ratios ρ_u, NUM vector
SNA	12	Additive NYM sensor noise variances V_y
SNR	13	Sensor noise ratios ρ_y, NYM vector
TH	14	Observational thresholds, a_i, NYM vector
ATTN	15	Attention allocations, f_i, NYM vector
CGAIN	16	Continuous control gains L_c, NUM by NTOT
DGAIN	17	Discrete conrol gains L_d, NUM by NTOT
INT	18	Transfer to subroutine MANNEW
PRINT	19	Printout interval for data

3.3 Subroutine UPDATE

The two major functions of this subroutine are to update the (time-varying) parameters in the human operator model, and to compute the discretized model equations. The parameters are updated using an alphanumeric code/index scheme. Parameters can be changed at time t via external or card inputs. They can also be changed periodically via an internal -- user supplied -- subroutine MANNEW. Table Ib defines the codes for UPDATE. The parameters are described in Sections 2.2 - 2.3.

The discretization of the human operator model equations follows the approach in Sections 2.4 - 2.5. A change in either A_m or B_m necessitates a recomputation of the discrete system matrices Φ_m and/or Γ_m. If continuous time feedback gains L_c are input, UPDATE computes the equivalent discretized gains $\hat{L}_d$ using the "average gain" method.

UPDATE initializes all of the man-model parameters to zero,(3) with the exception of A_m, B_m, C_m, L_d or L_c which must be input at time t_o. The subroutine can equate any of the first seven code parameters to their linear system counterparts.

3.4 Subroutine MAN

This is the major computational subroutine in the MCARLO program. It performs all of the human model computations summarized in Section 2.7. Thus, the basic function of MAN is to output (at time t) the NU control u to apply to the system over (t, t+Δ]. The dynamic inputs to MAN include the NY observations y(t), and the value of the deterministic input z over (t, t+Δ]. This latter requirement is expected to be relaxed through future modeling efforts.

3.5 Subroutine INFORM

This subroutine is used to store data for subsequent printing and/or plotting. Data is stored on disk files every (NPRNT)Δ seconds starting at $t=t_o$. For convenience, printed and plotted variables are stored on separate

---

(3) Initial $f_i = 1$

files. The value of NPRNT is an input parameter to MAIN, but can be changed via either system or update. For the system, any component of x, y or u may be output as data. For the man-model, any component of $\underline{\hat{x}}$, $\hat{y}$ or ν may be output where $\hat{y} = C\underline{\hat{x}}$.

3.6 Subroutine PRINTR

This routine, called the first time $t \geq t_F$, outputs the stored time histories of the selected variables. For plotted variables, an automatic scaling feature is used.

3.7 Program Operation

MCARLO has been designed to generate Monte-Carlo time histories of the signals in a man-machine conrol task. Each run of MCARLO generates one sample path. To obtain more elements in the ensemble it is necessary to make additional computer runs, using different values for the random number generator seed. The sample waveforms can all be stored for later ensemble averaging. As the number of samples $N_s \to \infty$, the sample statistics should converge to the ensemble (covariance) statistics that are computed by TIVAR.

4. INPUT DECK SETUP

There are three sections of input data for MCARLO as discussed below. In addition, there are three user-written subroutines: SYSNEW, MANNEW, and FDET.

4.1 Control Cards

There are 5 major control cards that are required by the MAIN program.

<u>Card 1</u> - Title Information

Column 1: blank

Columns 2-80: alphanumeric title information

<u>Card 2</u> - Random Number Seed, I10 Format

Field 1: IXYZ = any integer

<u>Card 3</u> - Time Information, 3E10.0, I10 Format

Field 1: DEL = discrete time step (sec)

Field 2: TO = initial time (sec)

Field 3: TF = final time (sec)

Field 4: NPRNT = printout frequency (integer)

<u>Card 4</u> - Print/Plot Information for System Variables, 3(20I1) Format

Field 1: Print/Plot codes for states 1 - NX

Field 2: Print/Plot codes for outputs 1 - NY

Field 3: Print/Plot codes for controls 1 - NU

<u>Card 5</u> - Print/Plot Information for Model Variables, 3(20I1) Format

Field 1: Print/Plot codes for state estimates 1 - NXM

Field 2: Print/Plot codes for output estimates 1 - NYM

Field 3: Print/Plot codes for KF residuals 1 - NYM

Note that on cards 4-5 each column of an associated field corresponds to one state, output estimate, etc. A single integer governs the printing or plotting of the time history of the variable:

0, or blank = no printing or plotting of the variable

1 = print time history vs. time

3 = plot time history vs. time

2 = print and plot time history vs. time

A maximum of 10 of any variables (e.g. states or outputs) can be printed on wide paper.

4.2 System Parameter Cards

These cards are used to change system parameters in the linear case. If the user supplies his own SYSTM subroutine, this section of data is omitted, or replaced by problem specific data cards.

Card 1 - Frequencies for internal time breaks, NDTS, 10I5 Format
The 10 fields are associated with the 10system parameter cards (see Table IIa) on a one-to-one basis. The I-th field is associated with Code I. NDTS(I) is the frequency (number of time steps) at which subroutine SYSNEW is to be called internally with KEY=I, starting at time T0. Calling SYSNEW with KEY=I sets ISFLAG(I)=1 for one time step. The actual parameter values must be changed internally by user-written code. If no code is supplied, the associated parameters retain their value.

Remaining Cards - These are used to change system parameters via external read-in at specified times. The deck setup follows a standard form.

Time Card - Cols. 1-4 Alphanumeric TIME
Cols. 11-20 Time of external break E10.0

Code Card - Cols. 1-5 One of the Alphanumeric codes in Table IIa
Cols. 8-10 Index NQQ for dimension information, I3

Parameter Cards - The new parameter values required by the code.

Table IIa: SYSTEM PARAMETER CARD INPUTS

CODE	KEY	INDEX	INPUT DATA	INITIAL VALUE
A	1	NX	$(A_s)_{ij}$ i=1,...,NX, j=1,...,NX	undef
B	2	NU	$(B_s)_{ij}$ i=1,...,NX, j=1,...,NU	undef
C	3	NY	$(C_s)_{ij}$ i=1,...,NY, j=1,...,NX	undef
D	4	NY	$(C_s)_{ij}$ i=1,...,NY, j=1,...,NU	0
E	5	NW	$(E_s)_{ij}$ i=1,...,NX, j=1,...,NW	0
F	6	NZ	$(F_s)_{ij}$ i=1,...,NX, j=1,...,NZ	0
WO	7	NW	$(W_s^o)_i$ i=1,...,NW	0
XINC	8	---	$(\delta x)_i$ i=1,...,NW	0
INT	9	KEY	---	---
PRINT	10	NPRNT	---	---

Table IIb: MAN-MODEL PARAMETER CARD INPUTS

CODE	KEY	INDEX	INPUT DATA	INITIAL VALUE
AM	1	NXM*	$(A_m)_{ij}$ i=1,...,NXM, j=1,...,NXM	undef
BM	2	NUM*	$(B_m)_{ij}$ i=1,...,NXM, j=1,...,NUM	undef
CM	3	NYM*	$(C_m)_{ij}$ i=1,...,NYM, j=1,...,NXM	undef
DM	4	NYM*	$(D_m)_{ij}$ i=1,...,NYM, j=1,...,NUM	0
EM	5	NWM*	$(E_m)_{ij}$ i=1,...,NXM, j=1,...,NWM	0
FM	6	NZM*	$(F_m)_{ij}$ i=1,...,NZM, j=1,...,NZM	0
WOM	7	NWM*	$(W_m^o)_i$ i=1,...,NWM	0
XHINC	8	---	$(\delta\hat{p})_i$ i=1,...,NXM	0
TD	9	---	τ	0
MNA	10	---	V_{ui} i=1,...,NUM	0
MNR	11	---	ρ_{ui} i=1,...,NUM in dB	- dB
SNA	12	---	V_{yi} i=1,...,NYM	0
SNR	13	---	ρ_{yi} i=1,...,NYM	- dB
TH	14	---	a_i i=1,...,NYM	0
ATTN	15	---	f_i i=1,...,NYM	1
CGAIN	16	---	$(L_c)_{ij}$ i=1,...,NUM, j=1,...,NXM+NUM	undef
DGAIN	17	---	$(L_d)_{ij}$ i=1,...,NUM< j=1,...,NXM+NUM	undef
INT	18	KEY	---	---
PRINT	19	NPRNT	---	---

*If <0, model parameters are automatically equated to system parameters, and no input data is needed.

The sequence of code card followed by new parameter values is repeated for all items that the user wishes to change at the given time. To change parameters at the next time, input a new time card, followed by a code card, input the parameter values, code card, etc. When using external (card) updates, the following rules must be observed:

1. $NX + NU \leq 15$
2. Time breaks must occur in increasing order.
3. Parameter cards should occur in the sequence listed in Table IIa. Thus, codes with lower KEY numbers should be read in first.
4. The parameter cards <u>must</u> be input immediately following the associated code card.
5. The last system parameter card must be an end-system card, containing the alphanumeric ENDS in cols. 1-4.
6. If they occur at the same time, external updates take precedence over internal updates.

The program reads all of the system update cards at the first time step, and stores all the information on a disk file for sequential read-in.

4.3 Man-Model Parameter Cards

These cards are used to change man-model parameters either internally in periodic mode, or via external card inputs. The deck setup is virtually identical in form to the previous section.

<u>Cards 1-2</u> - Frequencies for internal time breaks, NDTM, 19I5 Format
The 19 fields, spread over 2 cards, are associated with the 19 man-model parameter codes (see Table IIb) on a one-to-one basis. NDTM(I) is the frequency (number of time steps) at which the user supplied subroutine MANNEW is to be called internally with KEY=I. The operation is similar to that for SYSNEW.

Remaining Cards - These are used to change man-model parameters via external read-in at specified times. The deck setup follows the standard form as in the previous section 4.2:

Time Card - Cols. 1-4 Alphanumeric TIME

Cols. 11-20 Time of external break E10.0

Code Card - Cols. 1-5 One of the Alphanumeric codes in Table IIb

Cols. 8-10 Index NQQ for dimension information

Parameter Cards - As required by the associated code.

The sequence of code card - parameter card is repeated for all items the user wishes to change at the given time. To change parameters at the next time, the above three part sequence is repeated. The following rules must be observed:

1. $NXM + NUM \leq 15$
   $NUM = NU$
   $NYM = NY$
   $NZM = NZ$ (or $NZM = 0$)

2. Time breaks must occur in increasing order.

3. Parameter codes should occur in the sequence listed in Table IIb. Thus, codes should be sorted for reading according to increasing KEY numbers.

4. The parameter cards must be input immediately following the associated code card, unless $NQQ \leq 0$ for codes 1-7.

5. The last man-model parameter card must be an end-man card, containing the alphanumeric ENDM in cols. 1-4.

6. If they occur at the same time, external updates take precedence over internal updates.

A useful option is included for any of codes 1-7. If the specified NQQ $\leq$ 0, man-model parameters are automatically set equal to the corresponding system parameters in the linear case. Also, no read-in of model parameters is done.

4.4 Entering Parameter Data

Data is entered on the parameter cards in 8E10.0 Format, i.e., in floating point fields of 10 columns with a maximum of 8 fields per card. The numbers may be either in fixed-point (decimal) form or in scientific (exponential) form with the exponent right justified in the field. Matrices are entered one row at a time. If a row contains more than 8 entries, continue on a second card for that row. A new row always begins on a new card. Vectors are entered in similar 8E10.0 format: the first entry in the first field, second entry in the second field, etc.

4.5 User Written Routines

The three user written routines are SYSNEW(KEY), MANNEW (KEY), and FDET(K,T). The purpose of the first two routines for changing system and man-model parameters has been discussed earlier. The function FDET(K,T) is used to generate the time history of the deterministic (bias) inputs $z_i(t)$. Thus, at time T, and for inputs K, K=1,...,NZ,

$$FDET(K,T) = Z_K(T)$$

The user must supply his own code for FDET.

5. SAMPLE PROBLEM

A sample problem illustrating many of the features of the MCARLO program is given in this section. A description of the problem is presented first, followed by a listing of the user written subroutines, the input data deck, and a listing of the output.

5.1 Sample Problem Description

This problem analyzes an AAA tracking task. The controller tracks the azimuth angle of a target which is executing a level fly-by. The key element illustrated by this problem is the use of the FDET function to generate the time history of the deterministic input, i.e. the azimuth trajectory of the target.

The controller is explicitly presented with a display of the azimuth sighting error, and is assumed to derive the corresponding error rate. His task is to minimize this error by controlling a set of rate-aided second-order sight dynamics, his control being a hand-crank.

The target being tracked is executing a constant speed straight and level fly-by of 44 sec duration. The range of the target at crossover, R_c, is 3000 ft, and the speed, V, is 733 ft/sec producing a maximum azimuth angular velocity of about 14 deg/sec at crossover. Initially, 22 sec before crossover, the target is 16,126 ft from the crossover point, its azimuth angular position is -79.46 deg, and its azimuth angular velocity is 0.4683 deg/sec.

The system states are defined as follows:

x_1 = target azimuth angular position (degrees)
x_2 = target azimuth angular velocity (deg/sec)
x_3 = sight azimuth angular position (degrees)
x_4 = sight azimuth angular velocity (deg/sec)
x_5 = integral of the control input

It is assumed that the controller employs a "constant velocity" model of the target position. Consequently, the state space equations for the first two states (the deterministic states) are:

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = z(t),$$

where z(t), the deterministic input is the azimuth angular <u>acceleration</u> of the target. Thus, z(t) is given by:

$$z(t) = -2(V/R_c)^2 \frac{D(t)/R_c}{(1 + (D(t)/R_c)^2)^2} \tag{47}$$

where

V = the speed of the target = 733 ft/sec

R_c = the range of the target at crossover = 3000 ft

D(t) = the distacne of the target from the crossover point

$= D_o + Vt = -16,126 + 733t$

The FDET function computes z(t) according to Eq. 47.

The transfer function relating the sight position to the control input is:

$$\frac{x_3(s)}{u(s)} = \frac{64(s+1)}{s(s^2+12s+64)} = (1+1/s)\frac{64}{s^2+12s+64} \tag{48}$$

Consequently, the state space equations for the last three states (the controllable states) are:

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = -64x_3(t) - 12x_4(t) + 64x_5(t) + 64u(t) \tag{49}$$

$$\dot{x}_5 = u(t)$$

The displayed outputs are the azimuth sighting error and error rate and are given by:

$$
\begin{aligned}
y_1(t) &= x_1(t) - x_3(t) \\
y_2(t) &= x_2(t) - x_4(t)
\end{aligned}
\tag{50}
$$

Regarding the human's inherent limitatons, the observation noise to signal ratio (SNR) and motor noise ratio (MNR) are set to the nominal values of -20dB and -25dB, respectively. The perceptual time delay (TD) is set to 0.20 sec. Observational thresholds are set at 0.05 deg for y_1 (corresponding to 1% of the field of view of the gunsight), and 0.025 deg for y_2 (corresponding to a nominal differential threshold for motion).

The control gains, CGAIN, computed by another program, were chosen by setting the cost on error to unity, and adjusting the cost on control rate to produce a neuro-motor lag T_N of 0.1 sec.

Finally, since the mean initial states are far from zero, the human's estimator requires some time to settle down. The presence of the human's time delay further compounds this problem. To provide an initializing transition period while the human's estimator settles down, the sample problem is started at t=-6.0 sec, at which time the target is 20,524 ft from the crossover point, its azimuth angular position is -81.684 deg and its azimuth angular velocity is 0.2928 deg/sec. During this initialization period, the human's time delay is set to zero, and printouts are suppressed. At t=0, however, the time delay is set to 0.20 sec, and the printout interval is set to 1 sec.

The states are incremented, XINC, so that the initial angular position and velocity of the target are correct, and the human's state estimates are incremented, XHINC, so that the initial error and error rate are zero.

5.2 User Written Subroutines for the Sample Problem

```
C        MCP - PROBLEM DEPENDENT SUBPROGRAMS FOR MCARLO
C        INCLUDES:
C          1 - FUNCTION FDET
C          2 - SUBROUTINE SYSNEW
C          3 - SUBROUTINE MANNEW

         FUNCTION FDET(NQT,T)
C        GENERATES DETERMINISTIC INPUT

         DATA
     1   X0, Y0, V0 /-16126.0, 3000.0, 733.0/,
     2   R /57.296/

         X=X0+V0*T
         A=X/Y0
         B=1.0+A*A
         C=(V0/Y0)/B
         D=-2.0*A*C*C
         FDET=R*D
         RETURN

         END


         SUBROUTINE SYSNEW(NQQ)
C        INTERNAL UPDATES TO THE SYSTEM

         RETURN
         END


         SUBROUTINE MANNEW(NQQ)
C        INTERNAL UPDATES TO THE MAN

         RETURN
         END
```

5.3 Input deck for the Sample Problem

```
P-I-D CONTROLLER.  MONTE-CARLO SIMULATION
      56315
       0.05        -6.0        44.0        120
22221                22                      2
10100                20                      20
    0     0     0     0     0     0     0     0     0     0
A         5
        0.0         1.0         0.0         0.0         0.0
        0.0         0.0         0.0         0.0         0.0
        0.0         0.0         0.0         1.0         0.0
        0.0         0.0       -64.0       -12.0        64.0
        0.0         0.0         0.0         0.0         0.0
B         1
        0.0
        0.0
        0.0
       64.0
        1.0
F         1
        0.0
        1.0
        0.0
        0.0
        0.0
C         2
        1.0         0.0        -1.0         0.0         0.0
        0.0         1.0         0.0        -1.0         0.0
D         2
        0.0
        0.0
XINC
    -81.684      0.2928     -81.684      0.2928         0.0
ENDS
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
    0     0     0     0
AM        0
BM        0
CM        0
DM        0
FM        0
XHINC
    -81.684      0.2928     -81.684      0.2928         0.0
TD
       0.00
CGAIN
     -15.39      -3.303       6.359      0.6404       9.034
      10.00
MNR
      -25.0
SNR
      -20.0       -20.0
TH
       0.05       0.025
TIME      0         0.0
TD
       0.20
PRINT    20
ENDM
```

5.4 Output listing for the Sample Problem

```
STARTING MCARLO
 2-Dec-76    14:20

P-I-D CONTROLLER.  MONTE-CARLO SIMULATION

RANDOM NUMBER SEED=          56315
INTEGRATION TIME STEP =        0.050
INITIAL  TIME =        -6.000
TERMINAL  TIME =       44.000
PRINTOUT FREQUENCY =     120
SYSTEM INTERNAL BREAKS  INDEX  CODE  NDT

SYSTEM EXTERNAL BREAK AT T=  -6.000     CODE  A        INDEX=   5
     0.000E-01    1.000E+00    0.000E-01    0.000E-01    0.000E-01
     0.000E-01    0.000E-01    0.000E-01    0.000E-01    0.000E-01
     0.000E-01    0.000E-01    0.000E-01    1.000E+00    0.000E-01
     0.000E-01    0.000E-01   -6.400E+01   -1.200E+01    6.400E+01
     0.000E-01    0.000E-01    0.000E-01    0.000E-01    0.000E-01

SYSTEM EXTERNAL BREAK AT T=  -6.000     CODE  B        INDEX=   1
     0.000E-01
     0.000E-01
     0.000E-01
     6.400E+01
     1.000E+00

SYSTEM EXTERNAL BREAK AT T=  -6.000     CODE  F        INDEX=   1
     0.000E-01
     1.000E+00
     0.000E-01
     0.000E-01
     0.000E-01

SYSTEM EXTERNAL BREAK AT T=  -6.000     CODE  C        INDEX=   2
     1.000E+00    0.000E-01   -1.000E+00    0.000E-01    0.000E-01
     0.000E-01    1.000E+00    0.000E-01   -1.000E+00    0.000E-01

SYSTEM EXTERNAL BREAK AT T=  -6.000     CODE  D        INDEX=   2
     0.000E-01
     0.000E-01

SYSTEM EXTERNAL BREAK AT T=  -6.000     CODE  XINC     INDEX=   0
    -8.168E+01    2.928E-01   -8.168E+01    2.928E-01    0.000E-01
HUMAN INTERNAL BREAKS  INDEX  CODE  NDT

HUMAN EXTERNAL BREAK AT T=  -6.000     CODE  AM       INDEX=   0
HUMAN MODEL = SYSTEM

HUMAN EXTERNAL BREAK AT T=  -6.000     CODE  BM       INDEX=   0
HUMAN MODEL = SYSTEM

HUMAN EXTERNAL BREAK AT T=  -6.000     CODE  CM       INDEX=   0
HUMAN MODEL = SYSTEM

HUMAN EXTERNAL BREAK AT T=  -6.000     CODE  DM       INDEX=   0
HUMAN MODEL = SYSTEM

HUMAN EXTERNAL BREAK AT T=  -6.000     CODE  FM       INDEX=   0
HUMAN MODEL = SYSTEM

HUMAN EXTERNAL BREAK AT T=  -6.000     CODE  XHINC    INDEX=   0
    -8.168E+01    2.928E-01   -8.168E+01    2.928E-01    0.000E-01
```

```
HUMAN EXTERNAL BREAK AT T=  -6.000    CODE  TD        INDEX=   0
    0.000E-01

HUMAN EXTERNAL BREAK AT T=  -6.000    CODE  CGAIN     INDEX=   0
   -1.539E+01   -3.303E+00    6.359E+00    6.404E-01    9.034E+00
    1.000E+01

HUMAN EXTERNAL BREAK AT T=  -6.000    CODE  MNR       INDEX=   0
   -2.500E+01

HUMAN EXTERNAL BREAK AT T=  -6.000    CODE  SNR       INDEX=   0
   -2.000E+01   -2.000E+01

HUMAN EXTERNAL BREAK AT T=  -6.000    CODE  TH        INDEX=   0
    5.000E-02    2.500E-02
EQUIVALENT DISCRETE GAINS GENERATED:
   -1.187E+01   -2.871E+00    4.078E+00    4.587E-01    7.791E+00
    8.741E+00

HUMAN EXTERNAL BREAK AT T=   0.000    CODE  TD        INDEX=   0
    2.000E-01

HUMAN EXTERNAL BREAK AT T=   0.000    CODE  PRINT     INDEX=  20
MEAN OF X VECTOR
   -9.974E+00    3.223E+00   -9.863E+00    3.199E+00   -1.080E+01
RMS VALUES OF X VECTOR
    6.586E+01    3.843E+00    6.585E+01    1.257E+01    6.407E+01
MEAN OF Y VECTOR
   -1.112E-01    2.335E-02
RMS VALUES OF Y VECTOR
    2.074E+00    1.195E+01
MEAN OF U VECTOR
    1.580E+00
RMS VALUES OF U VECTOR
    9.259E+00
```

TIME	STATE 1	STATE 2	STATE 3	STATE 4	STATE 5
0.00	-7.947E+01	4.677E-01	-7.960E+01	4.125E-01	-7.974E+01
1.00	-7.898E+01	5.116E-01	-7.915E+01	9.271E-01	-7.943E+01
2.00	-7.844E+01	5.618E-01	-7.845E+01	5.847E-01	-7.888E+01
3.00	-7.786E+01	6.197E-01	-7.789E+01	7.591E-01	-7.833E+01
4.00	-7.721E+01	6.868E-01	-7.726E+01	7.768E-01	-7.773E+01
5.00	-7.648E+01	7.653E-01	-7.651E+01	8.245E-01	-7.707E+01
6.00	-7.567E+01	8.577E-01	-7.565E+01	8.815E-01	-7.629E+01
7.00	-7.477E+01	9.675E-01	-7.480E+01	7.255E-01	-7.546E+01
8.00	-7.374E+01	1.099E+00	-7.383E+01	1.253E+00	-7.458E+01
9.00	-7.257E+01	1.258E+00	-7.258E+01	1.210E+00	-7.349E+01
10.00	-7.122E+01	1.454E+00	-7.121E+01	1.464E+00	-7.227E+01
11.00	-6.965E+01	1.696E+00	-6.964E+01	1.852E+00	-7.086E+01
12.00	-6.782E+01	2.000E+00	-6.780E+01	1.842E+00	-6.921E+01
13.00	-6.564E+01	2.388E+00	-6.563E+01	2.699E+00	-6.729E+01
14.00	-6.303E+01	2.890E+00	-6.297E+01	2.853E+00	-6.499E+01
15.00	-5.984E+01	3.548E+00	-6.001E+01	4.086E+00	-6.228E+01
16.00	-5.590E+01	4.421E+00	-5.591E+01	4.590E+00	-5.887E+01
17.00	-5.095E+01	5.584E+00	-5.114E+01	5.437E+00	-5.472E+01
18.00	-4.467E+01	7.117E+00	-4.506E+01	7.672E+00	-4.956E+01
19.00	-3.667E+01	9.054E+00	-3.690E+01	8.665E+00	-4.282E+01
20.00	-2.658E+01	1.125E+01	-2.679E+01	1.159E+01	-3.428E+01
21.00	-1.436E+01	1.317E+01	-1.511E+01	1.323E+01	-2.380E+01
22.00	-6.721E-01	1.400E+01	-1.157E+00	1.569E+01	-1.150E+01
23.00	1.310E+01	1.325E+01	1.343E+01	1.394E+01	2.197E+00
24.00	2.550E+01	1.136E+01	2.582E+01	1.036E+01	1.535E+01
25.00	3.581E+01	9.160E+00	3.607E+01	8.595E+00	2.708E+01
26.00	4.401E+01	7.204E+00	4.423E+01	6.746E+00	3.684E+01
27.00	5.045E+01	5.651E+00	5.116E+01	4.902E+00	4.485E+01
28.00	5.551E+01	4.471E+00	5.603E+01	4.033E+00	5.108E+01
29.00	5.954E+01	3.586E+00	5.972E+01	3.544E+00	5.599E+01
30.00	6.279E+01	2.919E+00	6.311E+01	3.580E+00	6.001E+01
31.00	6.546E+01	2.410E+00	6.549E+01	2.432E+00	6.314E+01
32.00	6.767E+01	2.018E+00	6.781E+01	1.735E+00	6.580E+01
33.00	6.954E+01	1.710E+00	6.959E+01	1.648E+00	6.796E+01
34.00	7.113E+01	1.465E+00	7.121E+01	1.437E+00	6.979E+01
35.00	7.250E+01	1.267E+00	7.265E+01	1.688E+00	7.136E+01
36.00	7.368E+01	1.106E+00	7.374E+01	1.194E+00	7.270E+01
37.00	7.472E+01	9.735E-01	7.479E+01	8.909E-01	7.388E+01
38.00	7.564E+01	8.627E-01	7.569E+01	8.773E-01	7.489E+01
39.00	7.646E+01	7.696E-01	7.647E+01	7.686E-01	7.579E+01
40.00	7.719E+01	6.905E-01	7.721E+01	6.342E-01	7.659E+01
41.00	7.785E+01	6.228E-01	7.787E+01	5.325E-01	7.731E+01
42.00	7.844E+01	5.645E-01	7.841E+01	5.063E-01	7.793E+01
43.00	7.898E+01	5.139E-01	7.896E+01	4.871E-01	7.850E+01

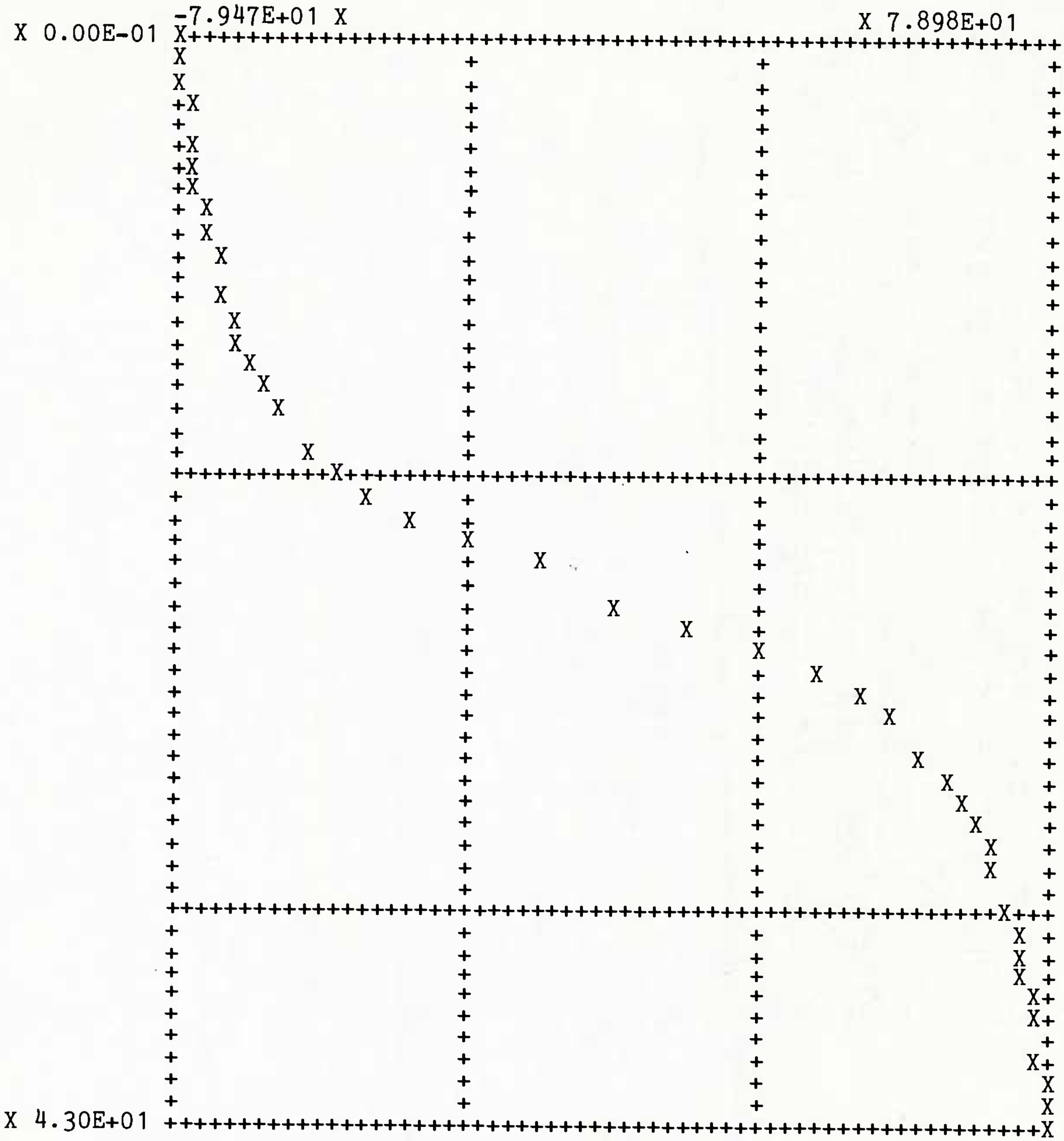
TIME	OUTPUT 1	OUTPUT 2
0.00	1.374E-01	5.519E-02
1.00	1.656E-01	-4.155E-01
2.00	2.328E-03	-2.293E-02
3.00	3.037E-02	-1.395E-01
4.00	5.052E-02	-9.001E-02
5.00	2.552E-02	-5.922E-02
6.00	-2.782E-02	-2.384E-02
7.00	3.871E-02	2.420E-01
8.00	8.750E-02	-1.538E-01
9.00	1.187E-02	4.863E-02
10.00	-3.470E-03	-9.822E-03
11.00	-1.513E-02	-1.561E-01
12.00	-1.551E-02	1.587E-01
13.00	-1.112E-02	-3.106E-01
14.00	-5.234E-02	3.662E-02
15.00	1.711E-01	-5.383E-01
16.00	1.513E-02	-1.693E-01
17.00	1.836E-01	1.468E-01
18.00	3.834E-01	-5.552E-01
19.00	2.285E-01	3.886E-01
20.00	2.084E-01	-3.483E-01
21.00	7.453E-01	-5.167E-02
22.00	4.852E-01	-1.686E+00
23.00	-3.329E-01	-6.948E-01
24.00	-3.137E-01	9.963E-01
25.00	-2.618E-01	5.649E-01
26.00	-2.212E-01	4.583E-01
27.00	-7.117E-01	7.491E-01
28.00	-5.173E-01	4.380E-01
29.00	-1.845E-01	4.247E-02
30.00	-3.179E-01	-6.614E-01
31.00	-3.509E-02	-2.130E-02
32.00	-1.338E-01	2.827E-01
33.00	-5.400E-02	6.138E-02
34.00	-7.917E-02	2.728E-02
35.00	-1.501E-01	-4.204E-01
36.00	-5.901E-02	-8.793E-02
37.00	-7.014E-02	8.256E-02
38.00	-4.147E-02	-1.456E-02
39.00	-1.335E-02	9.882E-04
40.00	-1.330E-02	5.629E-02
41.00	-1.702E-02	9.026E-02
42.00	3.559E-02	5.817E-02
43.00	2.873E-02	2.686E-02

TIME	CONTROL 1
0.00	1.485E-01
1.00	4.880E-01
2.00	5.266E-01
3.00	6.098E-01
4.00	6.487E-01
5.00	7.193E-01
6.00	8.189E-01
7.00	8.221E-01
8.00	1.016E+00
9.00	1.137E+00
10.00	1.303E+00
11.00	1.553E+00
12.00	1.752E+00
13.00	2.160E+00
14.00	2.563E+00
15.00	3.119E+00
16.00	3.814E+00
17.00	4.742E+00
18.00	5.922E+00
19.00	7.472E+00
20.00	9.740E+00
21.00	1.151E+01
22.00	1.352E+01
23.00	1.364E+01
24.00	1.246E+01
25.00	1.050E+01
26.00	8.561E+00
27.00	6.933E+00
28.00	5.321E+00
29.00	4.479E+00
30.00	3.609E+00
31.00	2.831E+00
32.00	2.292E+00
33.00	1.939E+00
34.00	1.693E+00
35.00	1.536E+00
36.00	1.241E+00
37.00	1.073E+00
38.00	9.634E-01
39.00	8.557E-01
40.00	7.458E-01
41.00	6.433E-01
42.00	5.878E-01
43.00	5.449E-01

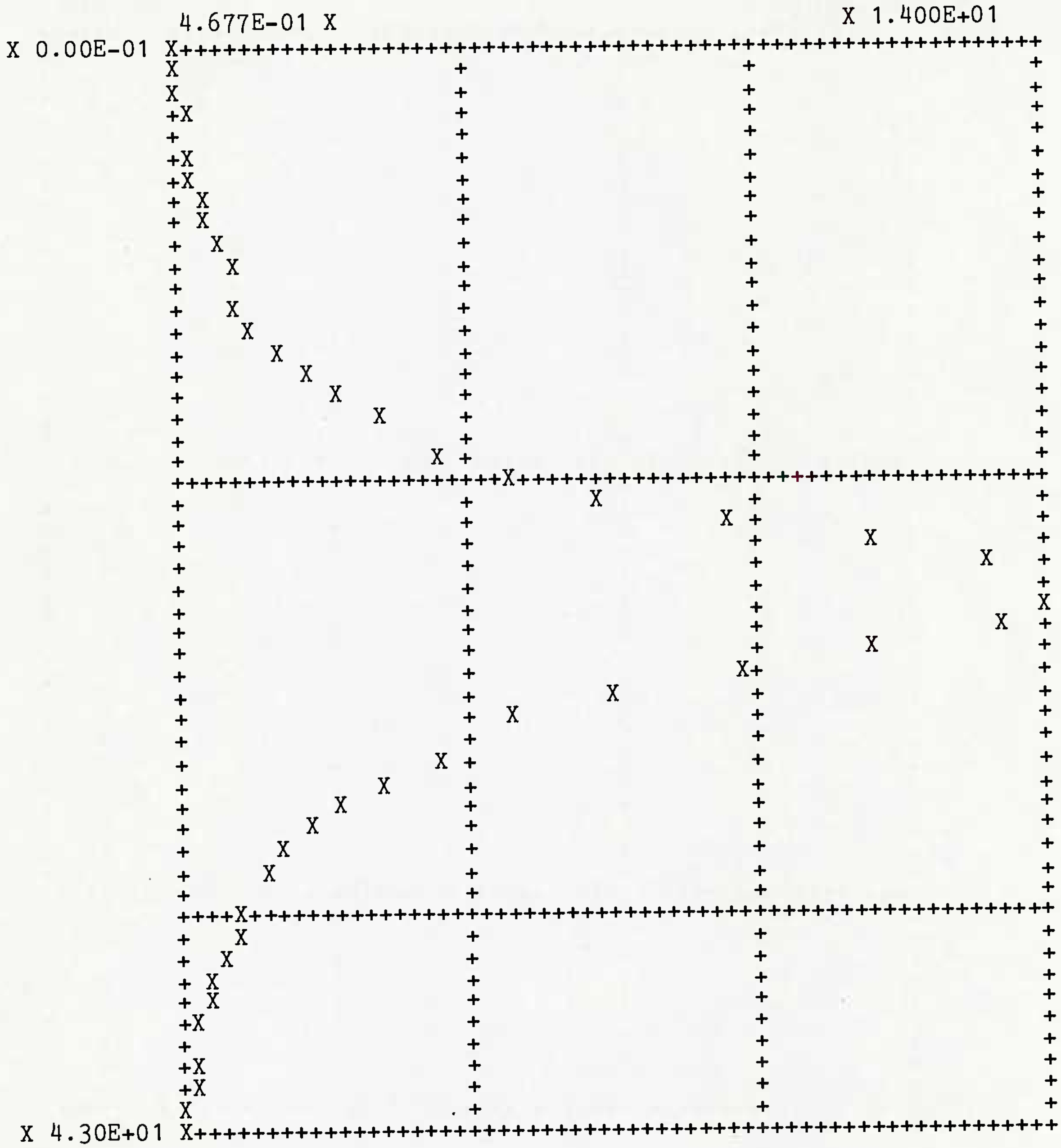
TIME	XMHAT 1	XMHAT 3
0.00	-7.975E+01	-7.980E+01
1.00	-7.911E+01	-7.924E+01
2.00	-7.847E+01	-7.848E+01
3.00	-7.782E+01	-7.786E+01
4.00	-7.718E+01	-7.723E+01
5.00	-7.646E+01	-7.648E+01
6.00	-7.559E+01	-7.561E+01
7.00	-7.478E+01	-7.477E+01
8.00	-7.372E+01	-7.380E+01
9.00	-7.256E+01	-7.257E+01
10.00	-7.122E+01	-7.120E+01
11.00	-6.962E+01	-6.965E+01
12.00	-6.784E+01	-6.782E+01
13.00	-6.560E+01	-6.568E+01
14.00	-6.290E+01	-6.296E+01
15.00	-5.981E+01	-5.998E+01
16.00	-5.584E+01	-5.590E+01
17.00	-5.095E+01	-5.109E+01
18.00	-4.481E+01	-4.512E+01
19.00	-3.670E+01	-3.685E+01
20.00	-2.652E+01	-2.672E+01
21.00	-1.464E+01	-1.523E+01
22.00	-6.236E-01	-1.045E+00
23.00	1.317E+01	1.346E+01
24.00	2.546E+01	2.585E+01
25.00	3.575E+01	3.610E+01
26.00	4.408E+01	4.419E+01
27.00	5.020E+01	5.110E+01
28.00	5.542E+01	5.599E+01
29.00	5.973E+01	5.994E+01
30.00	6.308E+01	6.326E+01
31.00	6.570E+01	6.571E+01
32.00	6.786E+01	6.801E+01
33.00	6.973E+01	6.974E+01
34.00	7.130E+01	7.137E+01
35.00	7.276E+01	7.279E+01
36.00	7.385E+01	7.387E+01
37.00	7.487E+01	7.491E+01
38.00	7.582E+01	7.579E+01
39.00	7.660E+01	7.657E+01
40.00	7.731E+01	7.731E+01
41.00	7.794E+01	7.797E+01
42.00	7.853E+01	7.852E+01
43.00	7.906E+01	7.906E+01

TIME	YMHAT 1
0.00	5.440E-02
1.00	1.286E-01
2.00	6.689E-03
3.00	3.072E-02
4.00	4.954E-02
5.00	1.996E-02
6.00	1.385E-02
7.00	-4.005E-03
8.00	7.715E-02
9.00	1.180E-02
10.00	-2.016E-02
11.00	2.993E-02
12.00	-2.005E-02
13.00	7.405E-02
14.00	6.200E-02
15.00	1.700E-01
16.00	5.668E-02
17.00	1.398E-01
18.00	3.162E-01
19.00	1.494E-01
20.00	2.030E-01
21.00	5.901E-01
22.00	4.217E-01
23.00	-2.861E-01
24.00	-3.876E-01
25.00	-3.435E-01
26.00	-1.058E-01
27.00	-9.002E-01
28.00	-5.654E-01
29.00	-2.010E-01
30.00	-1.746E-01
31.00	-1.552E-02
32.00	-1.482E-01
33.00	-3.513E-03
34.00	-6.806E-02
35.00	-2.637E-02
36.00	-2.666E-02
37.00	-3.475E-02
38.00	3.157E-02
39.00	2.213E-02
40.00	1.101E-03
41.00	-2.961E-02
42.00	1.100E-02
43.00	1.246E-03

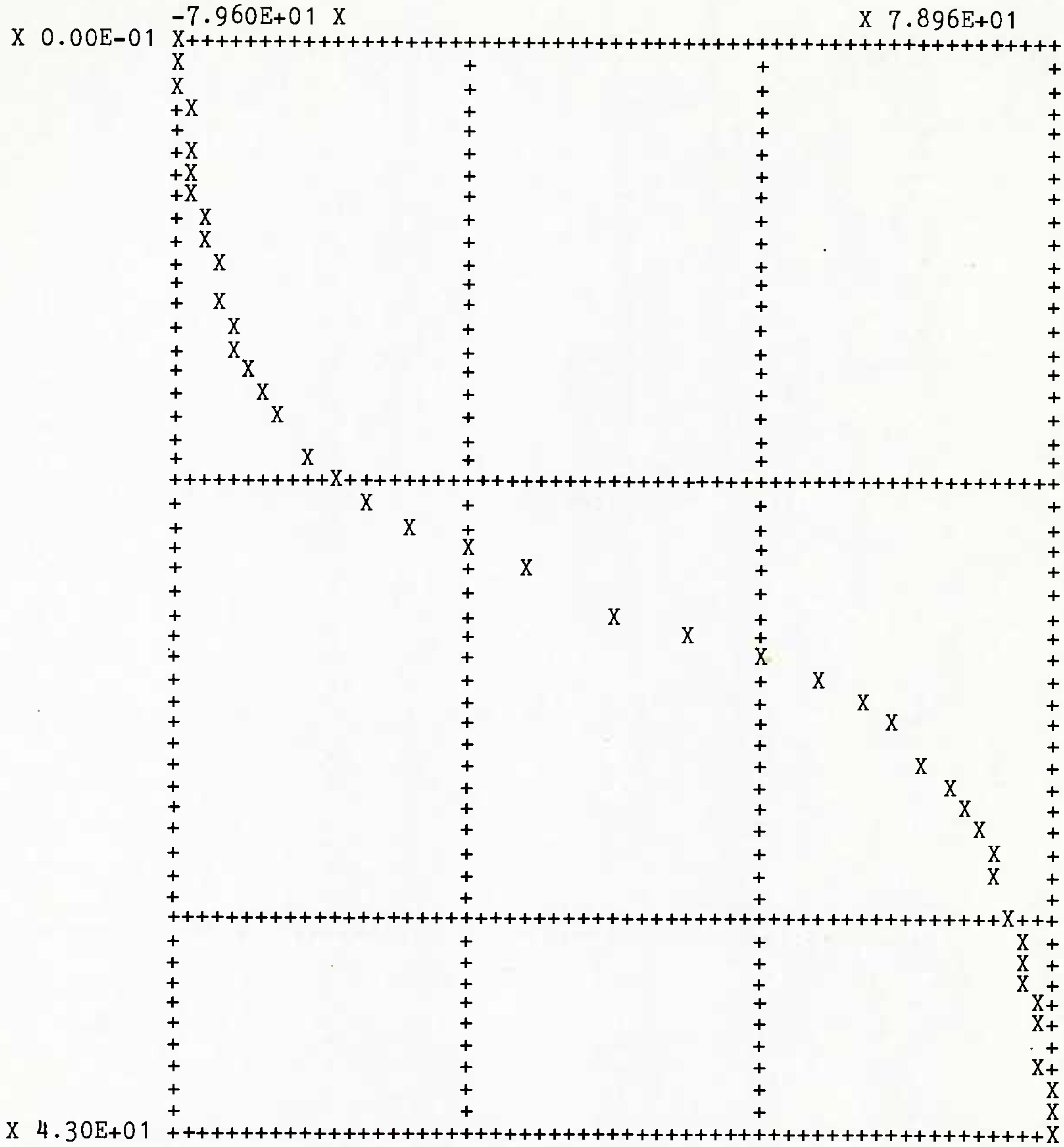
TIME	INOVAT 1
0.00	4.335E+03
1.00	8.887E-02
2.00	-2.526E+02
3.00	-6.851E-02
4.00	-2.569E-01
5.00	1.350E-01
6.00	2.387E-02
7.00	-7.828E-01
8.00	2.523E-01
9.00	1.157E+03
10.00	-9.278E+00
11.00	3.482E-01
12.00	2.497E-02
13.00	-2.527E-01
14.00	8.503E+02
15.00	-1.960E-01
16.00	2.483E-01
17.00	-3.536E-02
18.00	3.922E-01
19.00	1.561E-01
20.00	-1.411E-01
21.00	6.618E-01
22.00	2.696E-01
23.00	-5.703E-02
24.00	-7.549E-01
25.00	-4.178E-01
26.00	2.654E-01
27.00	-1.585E-01
28.00	2.572E-01
29.00	-1.941E-01
30.00	-1.921E-01
31.00	7.584E-02
32.00	-3.820E-02
33.00	-9.707E-02
34.00	-1.796E-01
35.00	1.117E-01
36.00	-2.200E-01
37.00	-5.386E-02
38.00	-5.220E-02
39.00	1.576E-01
40.00	3.340E-01
41.00	-9.230E-02
42.00	1.300E-01
43.00	-1.188E+00



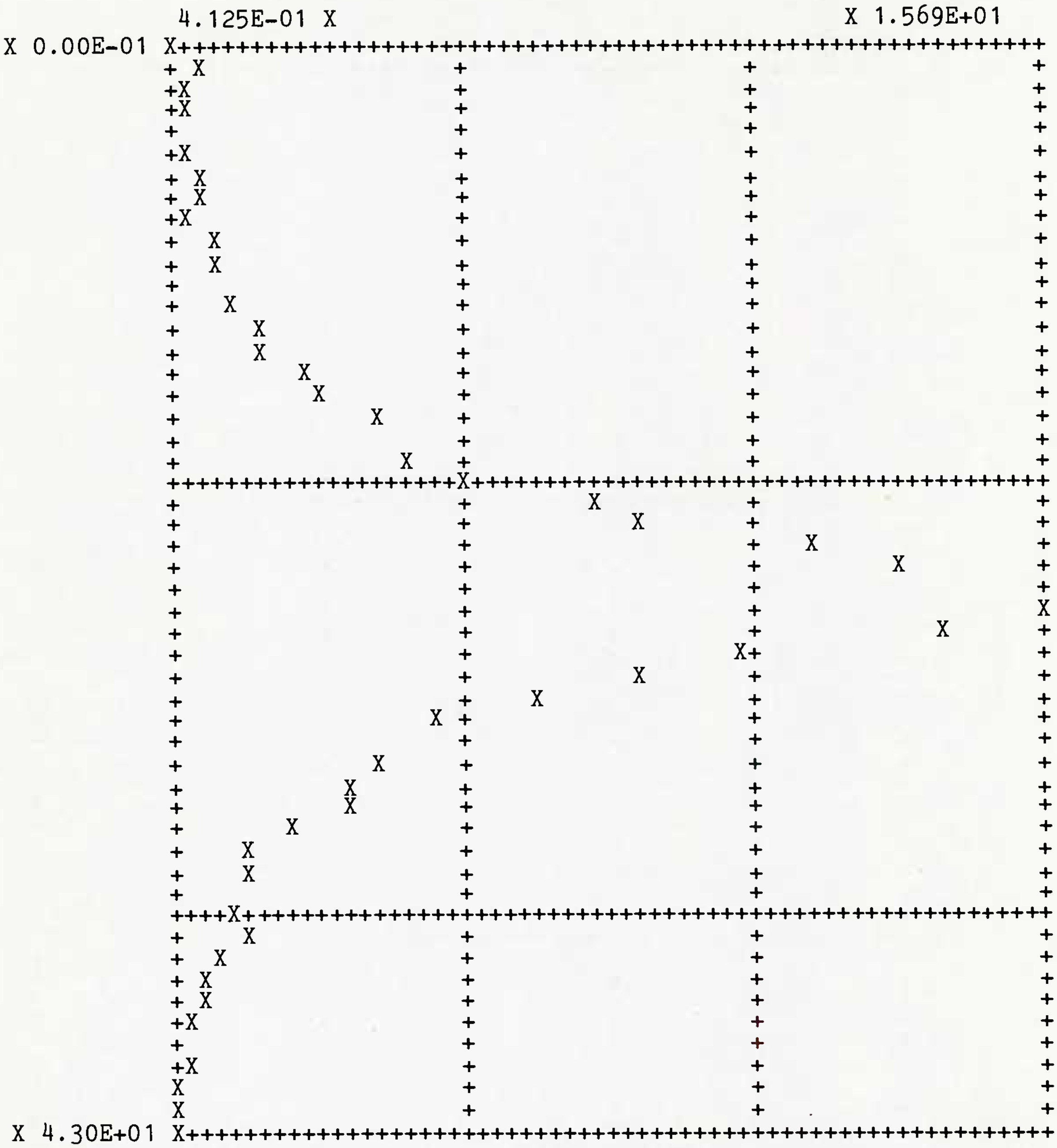
STATE 1



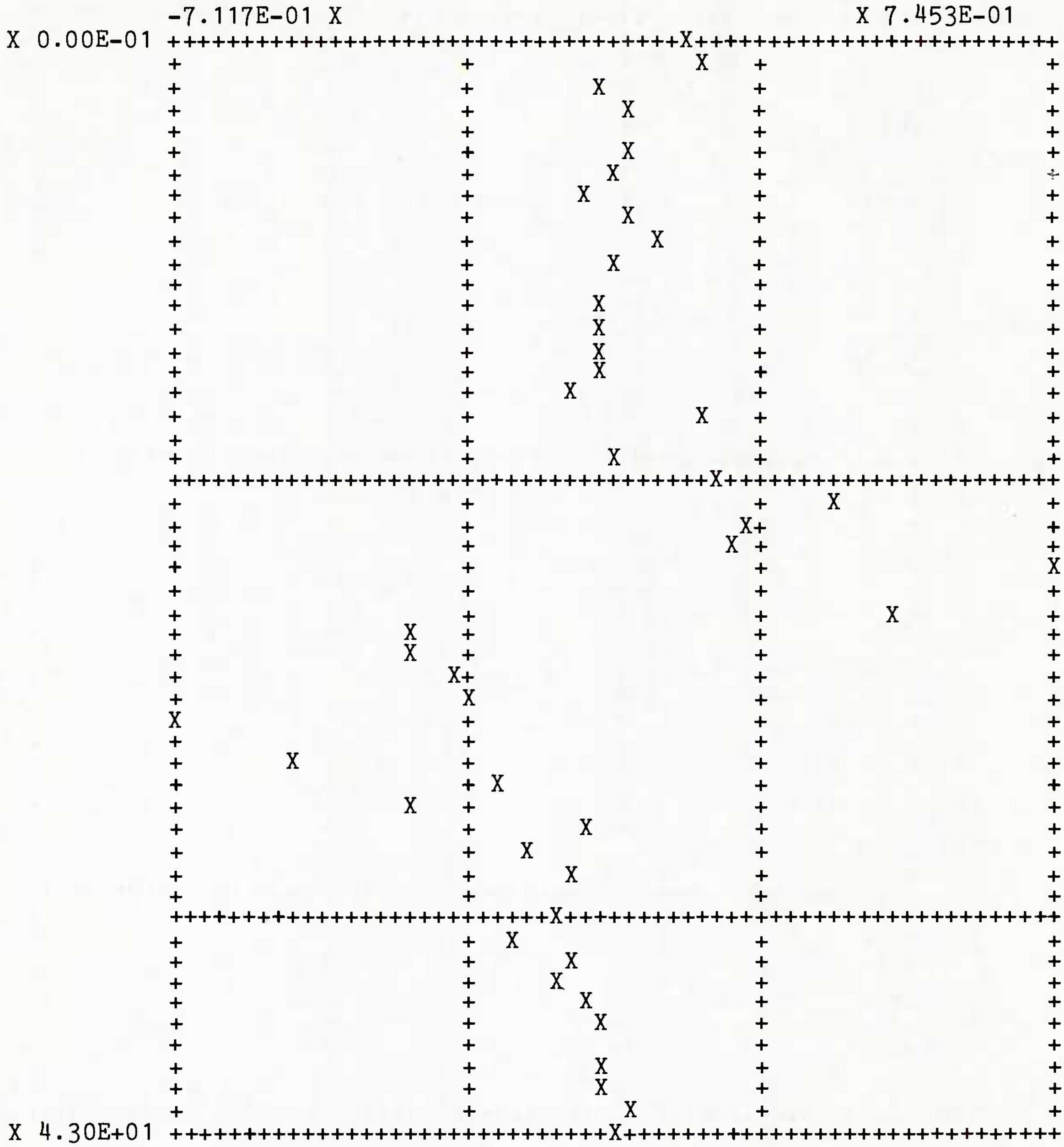
STATE 2



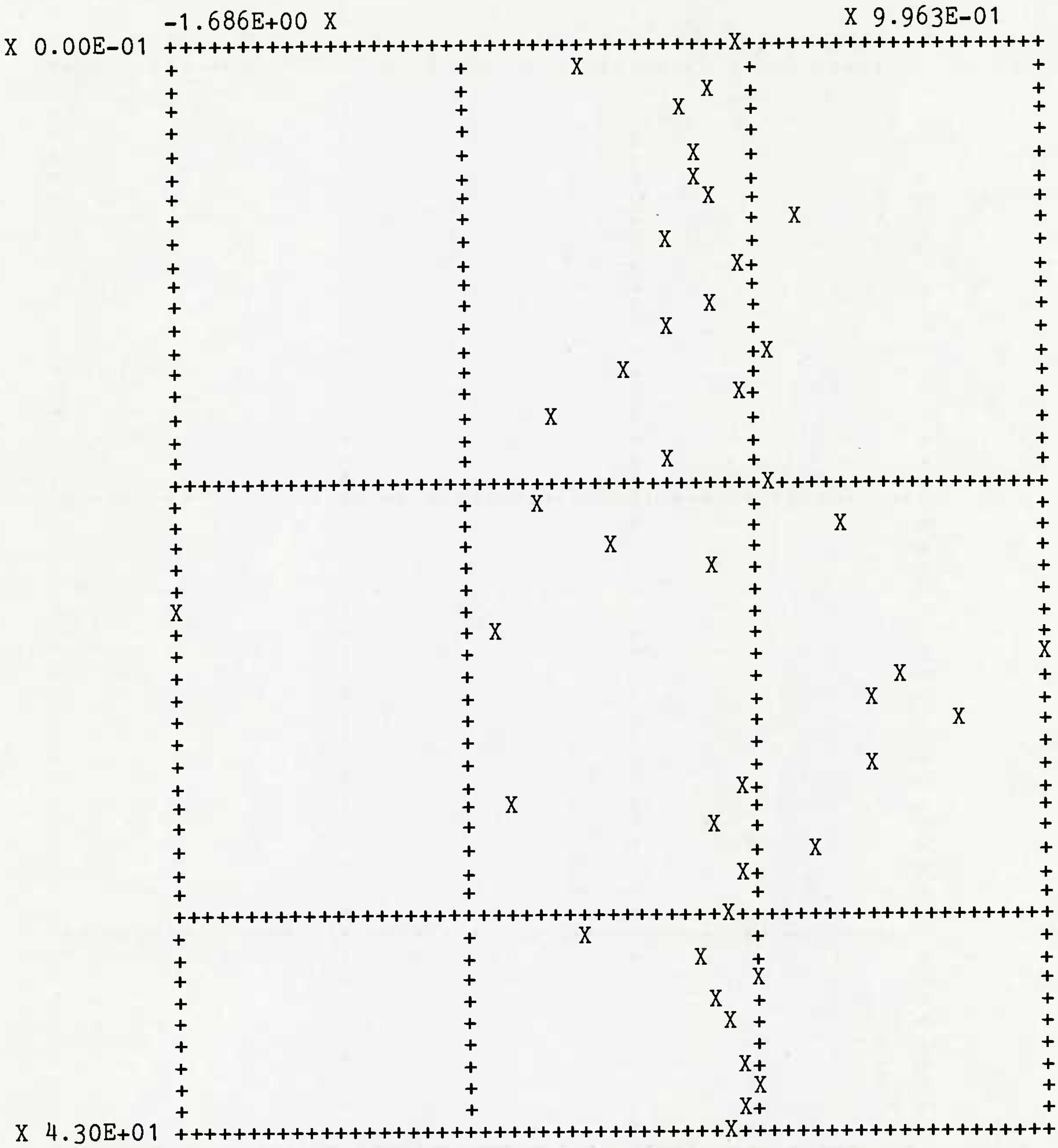
STATE 3



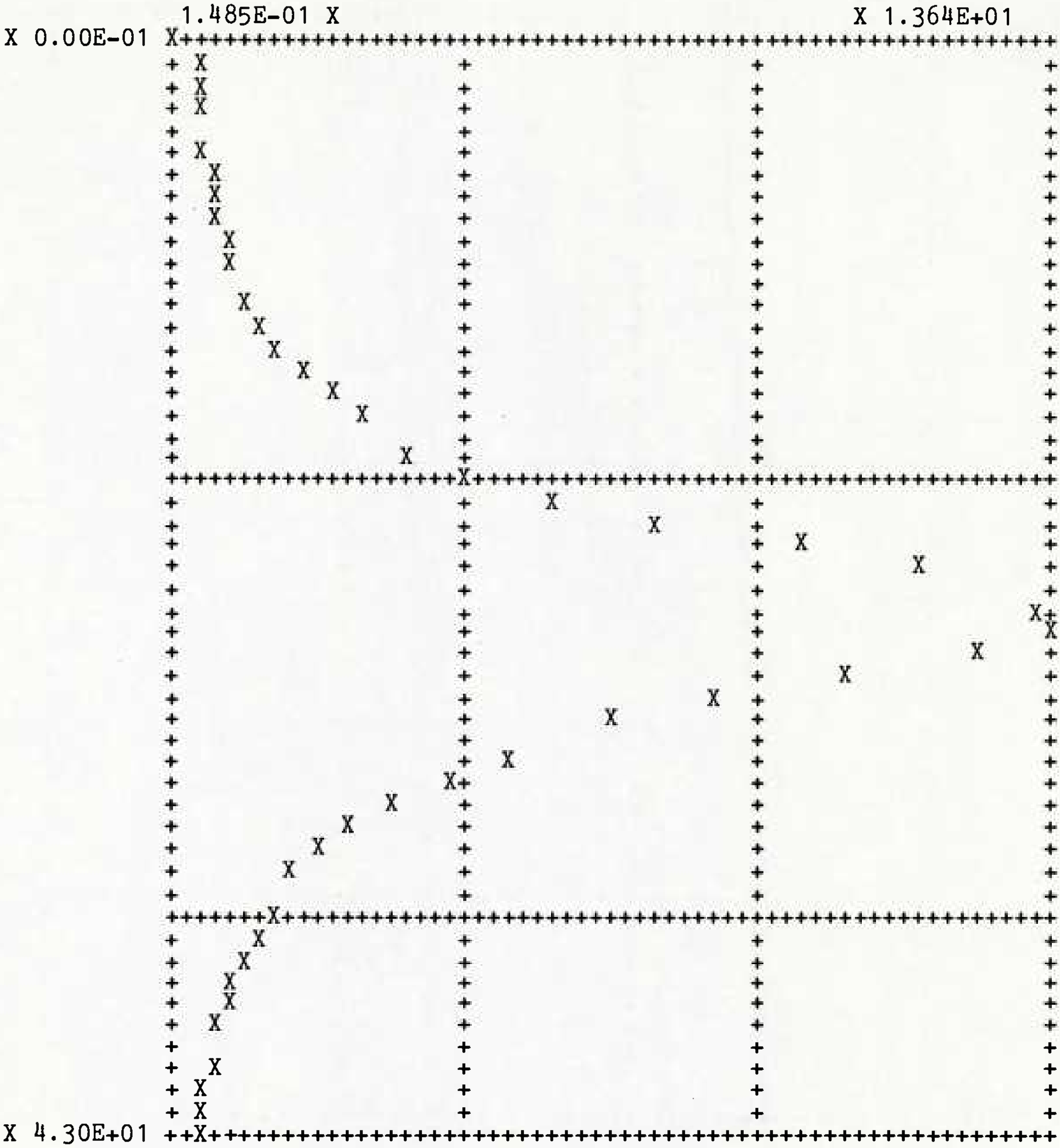
STATE 4



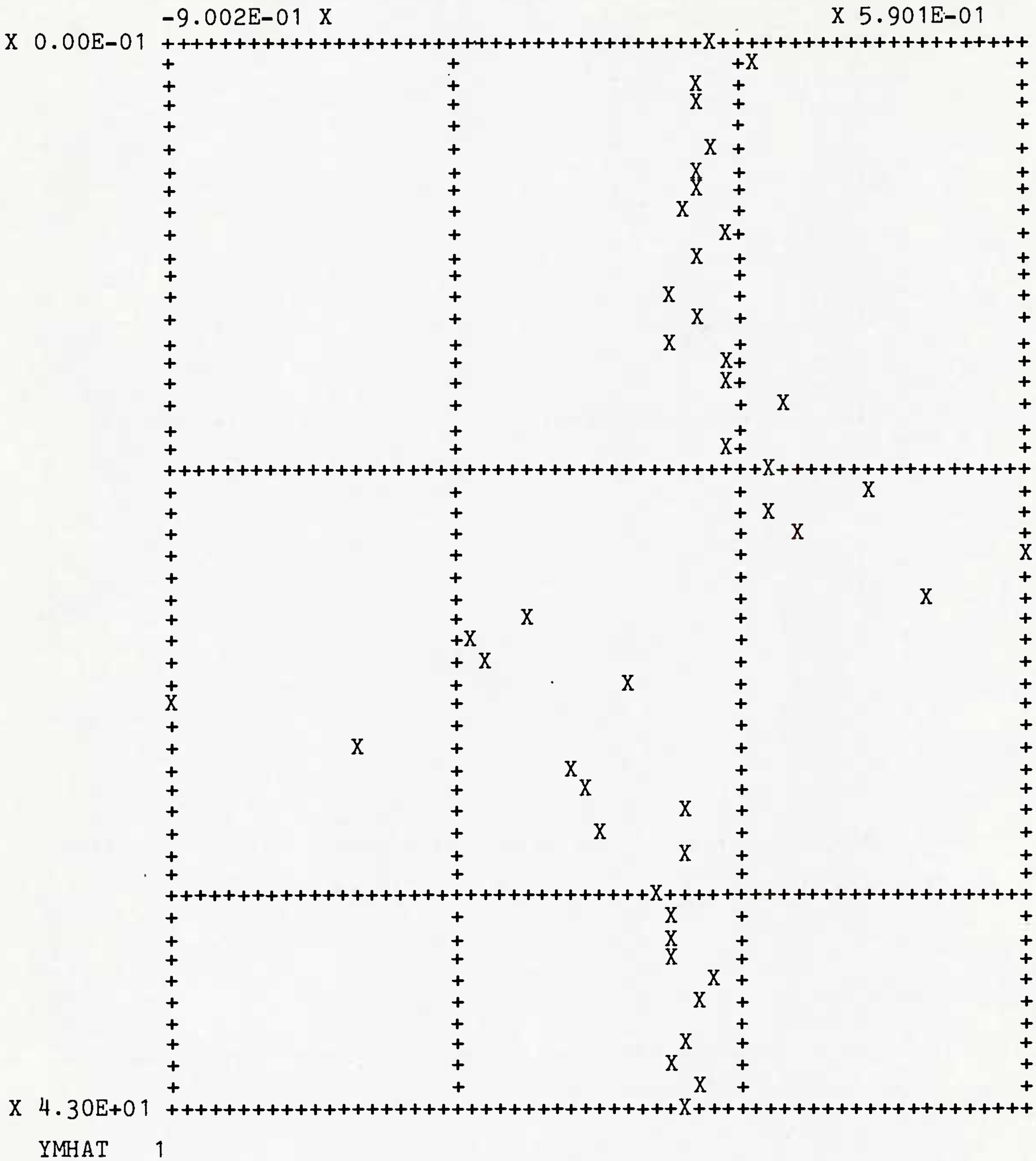
OUTPUT 1

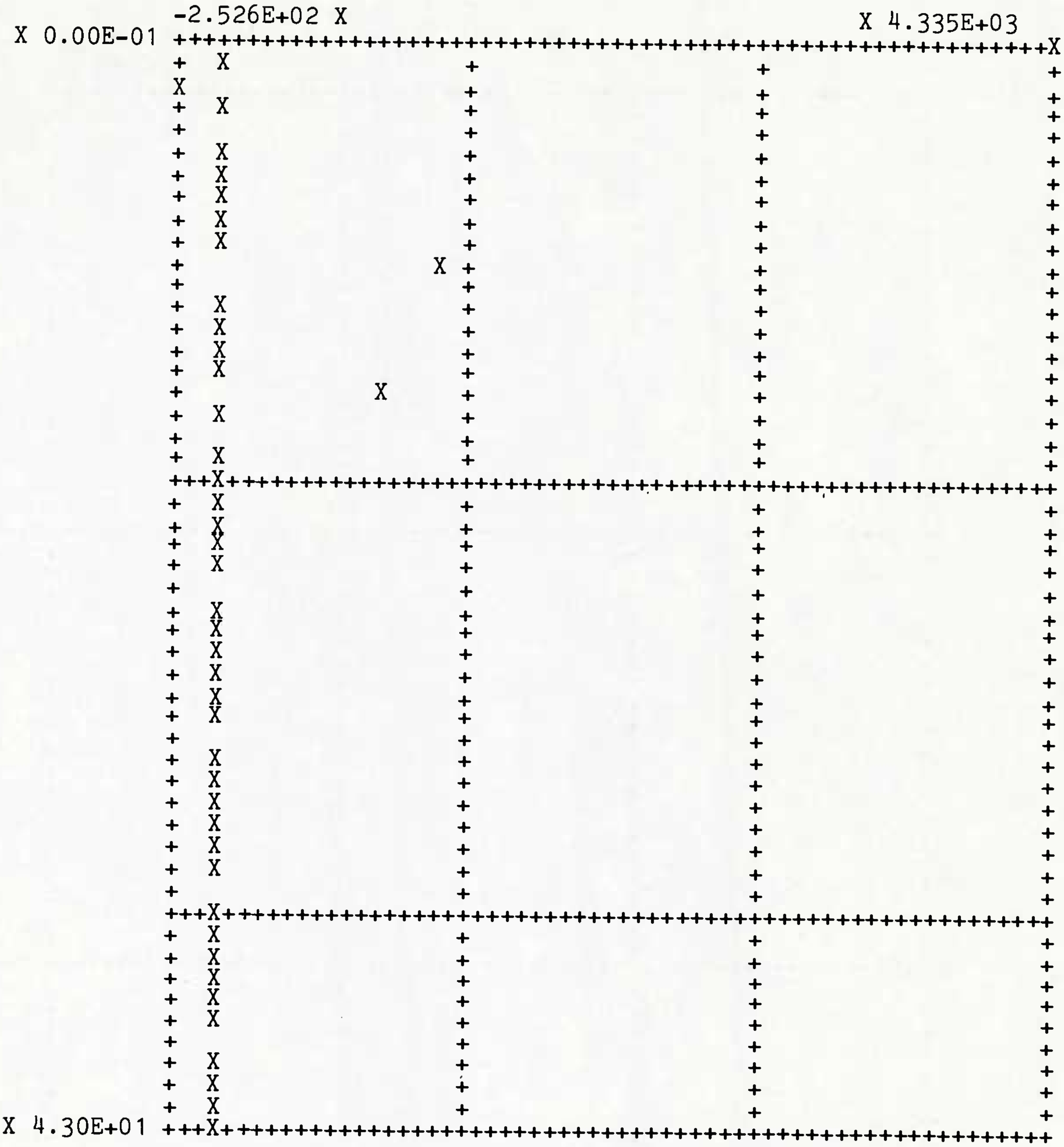


OUTPUT 2



CONTROL 1


-9.002E-01 X
X 5.901E-01
X 0.00E-01
X 4.30E+01
YMHAT 1




```
 INOVAT    1
STOPPING MCARLO
```

6. COMMON BLOCK USAGE

Named COMMON blocks are used to store most data arrays and to pass information among the various subroutines. These are described below.

1. /PLOT1/

   Required inputs for lineprint plot subroutine

2. /INOU/ KIN, KOUT, KPTR, KPUNCH, KDISK, IPOS, IGOS

   Logical unit numbers for I-O devices and disks for storage of system parameters and generated data.

3. /INFO/

   Storage of print/plot information including number of variables, min and max values for plot scaling, etc.

4. /TIMES/ TIME, DEL, TO, TF, NPRNT

   Time information t, Δ, t_o, t_f, printout frequency

5. /MAIN1/ NDIM, NDIM1, COM1 /MAIN2/ COM2

   Common blocks required for library subroutines

6. /SYSX/ NX, NU, BS, AS

   Linear System state parameters $BS=B_s$, $AS=A_s$

7. /SYSAD/ BD, AD

   Discrete System parameters $BD=\Gamma_s$, $AD=\Phi_s$

8. /SYSY/ NY, DS, CS

   Linear System output parameters $DS=D_s$, $CS=C_s$

9. /SYSW/ NW, WO, ES /SYSZ/ NZ, FS

   External System input parameters $WO=Wo_s$, $ES=E_s$, $FS=F_s$

10. /SYSINC/ XINC

    State increment δx

11. /MANX/ NXM, NUM, BM, AM

    Man-model state parameters $BM=B_m$, $AM=A_m$

12. /MANAD/ BD, AD

    Augmented discretized Man-model parameters BD=Γ, AD=Φ

13. /MANY/ NYM, DM, CM

    Man-model output parameters DM=D_m, CM=C_m (augmented)

14. /MANW/ NWM, WOM, EM /MANZ/ NZM, FM

    Man-model external input parameters WOM=W_m^o, EM=E_m, FM=F_m

15. /RATIOS/ PU, VU, PY, VY, TH, ATTN, SIGMA

    Model parameters ρ_u, V_u, ρ_y, V_y, a, f, $\Sigma_{k|k-1}$

16. /MANINC/ TD, NPRED, XHINC

    Man-model parameters TD=τ, NPRED=$[\tau/\Delta]$=N, XHINC=$\delta\hat{p}$

17. /GAINBK/ CGN

    CGN = discrete control gains L_d or equivalent discrete gains $\hat{L}_d$ computed from L_c

7. MCARLO LISTING

```
C       NO TABS

C       MCARLO - TIME VARYING MONTE CARLO MAN/MACHINE SIMULATION
C       INCLUDES
C        1 - BLOCK DATA MCDAT    - INITIALIZES VARIOUS COMMON BLOCKS
C        2 - MAIN.               - CALLS SUBROUTINE MCARLO
C        3 - SUBROUTINE MCARLO   - PRIMARY SUBPROGRAM

C       ALSO REQUIRES THE FOLLOWING SUBPROGRAM FILES

C       MCCMP - COMPUTATIONS FOR MCARLO
C       INCLUDES
C        1 - SUBROUTINE SYSTM   - PROPAGATES THE SYSTEM'S RESPONSE
C        2 - SUBROUTINE MAN     - PROPAGATES THE MAN'S RESPONSE
C        3 - FUNCTION GAUSS     - PICKS A NUMBER FROM A GAUSSIAN
C                                 DISTRIBUTION

C       MCP - PROBLEM DEPENDENT SUBPROGRAMS FOR MCARLO
C       INCLUDES
C        1 - SUBROUTINE FDET     - GENERATES DETERMINISTIC INPUT
C        2 - SUBROUTINE SYSNEW   - INTERNAL UPDATES TO THE SYSTEM
C        3 - SUBROUTINE MANNEW   - INTERNAL UPDATES TO THE MAN

C       MCIO - I/O FOR MCARLO
C       INCLUDES
C        1 - SUBROUTINE UPDATE   - PERFORM EXTERNAL UPDATES
C        2 - SUBROUTINE INFORM   - DO OUTPUT FOR A SINGLE TIME STEP
C        3 - SUBROUTINE PUTOUT   - SAVES OUTPUT ON FILES
C        4 - SUBROUTINE PRINTR   - PRINT THE OUTPUT AT THE END OF A RUN

C       KPLOT - LINEPRINTER PLOTTING PACKAGE
C       INCLUDES
C        1 - SUBROUTINE KPLOT
C        2 - SUBROUTINE ADJUST
C        3 - SUBROUTINE QINIT
C        4 - SUBROUTINE KPLOTC
C        5 - SUBROUTINE QPLOT
C        6 - SUBROUTINE PLACE
C        7 - SUBROUTINE QPRINT
C
        BLOCK DATA MCDAT
C       INITIALIZES VARIOUS COMMON BLOCKS

        COMMON
     1  /INOU/   KIN, KOUT, KPTR, KPUNCH,
     1           KDISK, IPOS, IGOS
     2  /PLOT1/  NV, NH, NCPW, LW, XL, XH, YL, YH, NXES, NDIR, IST,
     2           NGLV, NGLH, BSYM, GSYM, PSYM, ND1, ND2, NOUT

        DATA
     1    NV,  NH, NXES, NDIR, NGLV, NGLH, BSYM, GSYM, PSYM, IST, ND1
     1  / 51, 101,    1,   10,   20,   20,  1H+,  1H+,  1HX,  11,   1/

        END
```

```
      PROGRAM MCMAIN
     1 (INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT,
     2  DISK, POS, GOS, TAPE7=DISK, TAPE8=POS, TAPE9=GOS)

C     MAIN PROGRAM
C     CALLS SUBROUTINE MCARLO

      CALL MCARLO
      END
```

```
      SUBROUTINE MCARLO
C     PRIMARY SUBPROGRAM

      DIMENSION
     1 TITLE(8),
     2 X(30), Y(30), U(30), W(30), Z(30),
     3 XMHAT(30), YMHAT(30), RES(30)

      COMMON
     1 /INOU/   KIN, KOUT, KPTR, KPUNCH,
     1          KDISK, IPOS, IGOS
     2 /PLOT1/  NV, NH, NCPW, LW, XL, XH, YL, YH, NXES, NDIR, IST,
     2          NGLV, NGLH, BSYM, GSYM, PSYM, ND1, ND2, NOUT
     3 /MAIN1/  NDIM, NDIM1, COM1(15,15)
     4 /MAIN2/  COM2(15,15), STORE(1800)
     5 /TIMES/  TIME, DEL, TO, TEND, NPRNT
     6 /INFO/   NREC, NPRINT, NPLOT, LPRNTS(60), LPRNTM(60),
     6          IP(6), IG(6), SMIN(21), SMAX(21)
     7 /FILES/  KKB, NAMIN, NAMOUT, NAMDYN, NAMPCH, NAMDSK,
     7          NAMPOS, NAMGOS
     Z /COMP1/  X, Y, U, W, Z, XMHAT, YMHAT, RES

      DATA
     1 KIN, KOUT, KPTR, NOUT /5, 6, 6, 6/,
     2 KPUNCH, KDISK, IPOS, IGOS /6, 7, 8, 9/


C     SET NDIM
      NDIM=15
      NDIM1=NDIM+1

C     WRITE DAYTIM
150   CALL PAGEFD(KOUT,1)
      WRITE (KOUT,1500)
1500  FORMAT (1H ,/,1H ,15HSTARTING MCARLO)
      CALL DAYTIM(KOUT)

C     ZERO THE VECTORS
200   DO 220 I=1,30
      X(I)=0.0
      Y(I)=0.0
      W(I)=0.0
      Z(I)=0.0
      U(I)=0.0
220   CONTINUE
      ISTEP=0

C     GET TITLE AND RANDOM NUMBER SEED
300   READ (KIN,1040) (TITLE(I), I=1,8)
      IF (EOF(KIN)) 1000,350
1040  FORMAT (8A10)
350   WRITE (KOUT,1045) (TITLE(I), I=1,8)
1045  FORMAT (/,1H ,8A10,/,1H )
      READ (KIN,1051) IDUM
1051  FORMAT (I10)
      WRITE (KOUT,1052) IDUM
1052  FORMAT (21H RANDOM NUMBER SEED= ,I10)
      CALL RANSET(IDUM)

C     SPECIFY DEL, TO, TEND, AND NPRNT
330   READ (KIN,1060) DEL, TO, TEND, NPRNT
1060  FORMAT (3E10.0,I10)
      TEND=IFIX((TEND-TO+0.0001)/DEL)*DEL+TO
      WRITE (KOUT,1065) DEL, TO, TEND, NPRNT
1065  FORMAT (25H INTEGRATION TIME STEP = ,F10.3,/,
     1 17H INITIAL TIME =  ,F10.3,/,
```

```
     2  17H TERMINAL TIME = ,F10.3,/,
     3  22H PRINTOUT FREQUENCY = ,I5)
C       IDENTIFY VARIABLES FOR OUTPUT
        READ (KIN,1070) LPRNTS
        READ (KIN,1070) LPRNTM
1070    FORMAT (60I1)
C       SET THE TIME AND ENTER THE MAIN COMPUTATIONAL LOOP
        TIME=TO
C       START BY UPDATING THE SYSTEM
600     CALL SYSTM(TIME,X,Y,U,W,Z)
C       RETURNS X,Y AT T,  GIVEN U,W,Z OVER (T-DEL,T)
C       UPDATE THE MAN
        CALL MAN(IDUM,Y,U,Z,XMHAT,YMHAT,RES)
C       RETURNS U OVER (T,T+DEL)  GIVEN Y AT T
C       CALL INFORM TO DO A PRINTOUT IF ONE IS DUE AT THIS TIME
        CALL INFORM(ISTEP,X,Y,U,XMHAT,YMHAT,RES)
C       UPDATE THE TIME
        ISTEP=ISTEP+1
        TIME=DEL*ISTEP+TO
C       IF THE TIME IS NOT EXPIRED DO ANOTHER ITERATION
        IF (TIME+0.0001 .GE. TEND) GO TO 800
        GO TO 600
C       TIME IS EXPIRED.  DO OUTPUT AND START AGAIN
800     CALL PRINTR(NPRINT,NPLOT,X,Z)
        GO TO 200
C       INPUT FILE IS EMPTY
C       WRITE MESSAGE AND EXIT
1000    WRITE (KOUT,9000)
9000    FORMAT (1H ,15HSTOPPING MCARLO)
        CALL EXIT

        END
```

```
C       MCCMP - COMPUTATIONS FOR MCARLO
C       INCLUDES
C        1 - SUBROUTINE SYSTM
C        2 - SUBROUTINE MAN
C        3 - FUNCTION GAUSS

        SUBROUTINE SYSTM(T,X,Y,U,W,Z)
C       INTEGRATES SYSTEM EQUATIONS ONE TIME STEP
C       HANDLES EXTERNAL AND INTERNAL SYSTEM UPDATES
C       THIS VERSION IS FOR A LINEAR SYSTEM USING TRANSITION MATRIX
C       PROPAGATION
C       A MORE GENERAL VERSION WOULD USE RUNGE-KUTTA.

        DIMENSION
     1  X(30), Y(30), U(30), W(30), Z(30),
     2  DUM(20), NSTEP(10), ISFLAG(16), AINT(15,15), MDUM(16),
     3  XMEAN(30), YMEAN(30), UMEAN(30),
     4  XRMS(30), YRMS(30), URMS(30)

        INTEGER
     1  SCODE(10)

        COMMON
     1  /INOU/   KIN, KOUT, KPTR, KPUNCH,
     1           KDISK
     5  /TIMES/  TIME, DEL, TO, TEND, NPRNT
     8  /SYSX/   NX, NU, BS(15,4), AS(15,15)
     9  /SYSAD/  BD(15,4), AD(15,15)
     A  /SYSY/   NY, DS(15,4), CS(15,15)
     B  /SYSW/   NW, WO(4), ES(15,4)
     C  /SYSZ/   NZ, FS(15,4)
     D  /SYSINC/ XINC(15)

        DATA
     1  NSCODE, LTIME, LEND
     1  /10, 4HTIME, 4HENDS/,
     2  SCODE
     2  /1HA, 1HB, 1HC, 1HD, 1HE,
     2   1HF, 2HWO, 4HXINC, 3HINT, 5HPRINT/

        IF (T .GT. TO+0.0001) GO TO 100

C       INITIALIZATION
C       SPECIFY THE SYSTEM INTERNAL BREAKS
        READ (KIN,1050) (NSTEP(I), I=1,NSCODE)
1050    FORMAT (16I5)
        WRITE (KOUT,1060)
1060    FORMAT (24H SYSTEM INTERNAL BREAKS ,17H INDEX  CODE  NDT)
        DO 65 I=1,NSCODE
        IF (NSTEP(I) .LE. 0) GO TO 65
        WRITE (KOUT,1070) I, SCODE(I), NSTEP(I)
1070    FORMAT (28X,I2,3X,A5,1X,I5)
65      CONTINUE

C       WRITE THE NEXT BATCH OF INPUT CARDS UNTIL AN 'END' CARD
        REWIND KDISK
72      READ (KIN,1075) MDUM
1075    FORMAT (16A5)
        WRITE (KDISK,1075) MDUM
        II=MDUM(1)
        IF (II.EQ.LEND) GO TO 80
        GO TO 72
80      TNEXT=TO
        REWIND KDISK
        KIN1=KIN
```

```
      IXYZ=35617
C     INITIALIZE SOME MORE QUANTITIES
      NW=0
      NZ=0
      DO 85 I=1,15
      XINC(I)=0.0
      DO 85 J=1,4
      DS(I,J)=0.0
85    CONTINUE
      RNPTS=DEL/(TEND-TO)
      DO 90 I=1,30
      XMEAN(I)=0.0
      YMEAN(I)=0.0
      UMEAN(I)=0.0
      XRMS(I)=0.0
      YRMS(I)=0.0
      URMS(I)=0.0
90    CONTINUE

C     TAKE CARE OF INTERNAL SYSTEM BREAKS
100   DO 105 I=1,NSCODE
      ISFLAG(I)=0
      IF (NSTEP(I) .EQ. 0) GO TO 105
      ITME=IFIX((T-TO+0.0001)/DEL)
      IF (MOD(ITME,NSTEP(I)) .EQ. 0) CALL SYSNEW(I)
105   CONTINUE

C     TAKE CARE OF EXTERNAL SYSTEM BREAKS
110   IF (T+0.0001 .LT. TNEXT) GO TO 500
120   READ (KDISK,1130) IDEN, NQQ, BRKT
1130  FORMAT (A5,2X,I3,E10.0)
      IF (IDEN.NE.LEND) GO TO 140
135   TNEXT=1.0E+05
      GO TO 110
140   IF (IDEN .NE. LTIME) GO TO 150
      TNEXT=BRKT
      GO TO 110

C     SEARCH THROUGH THE UPDATE CODES, SCODE(KEY)
150   DO 160 KEY=1,NSCODE
      IF (IDEN.EQ.SCODE(KEY)) GO TO 170
160   CONTINUE

C     CODE WAS ILLEGAL
      WRITE (KOUT,1165) IDEN
1165  FORMAT (23H ILLEGAL INPUT CODE OF ,A5)
      CALL EXIT

C     DO THE SPECIFIED SYSTEM EXTERNAL UPDATE
170   ISFLAG(KEY)=1
      IO=2
      KIN=KDISK
      WRITE (KOUT,1175) TIME, IDEN, NQQ
1175  FORMAT (/,28H SYSTEM EXTERNAL BREAK AT T= ,F8.3,4X,4HCODE,2X,A5,
     1 4X,7HINDEX= ,I3)
      GO TO (1,2,3,4,5,6,7,8,9,10), KEY

C     SYSTEM DYNAMICS - A, B, C, D, E
1     NX=NQQ
      CALL MATIO(AS,NX,NX,IO)
      GO TO 120
2     NU=NQQ
      CALL MATIO(BS,NX,NU,IO)
      GO TO 120
3     NY=NQQ
      CALL MATIO(CS,NY,NX,IO)
```

```
      GO TO 120
4     NY=NQQ
      CALL MATIO(DS,NY,NU,IO)
      GO TO 120
5     NW=NQQ
      IF (NW.GT.0) CALL MATIO(ES,NX,NW,IO)
      GO TO 120

C     DETERMINISTIC INPUT (F MATRIX) - F
6     NZ=NQQ
      IF (NZ.GT.0) CALL MATIO(FS,NX,NZ,IO)
      GO TO 120

C     DRIVING NOISE - WO
7     NW=NQQ
      IF (NW.GT.0) CALL VECTIO(WO,NW,IO)
      GO TO 120

C     INCREMENT TO STATES - XINC
8     CALL VECTIO(XINC,NX,IO)
      GO TO 120

C     CALL AN INTERNAL UPDATE - INT
9     CALL SYSNEW(NQQ)
      GO TO 120

C     SET PRINT INTERVAL - PRINT
10    NPRNT=NQQ
      GO TO 120

C     NO MORE SYSTEM EXTERNAL UPDATES AT THIS TIME
500   KIN=KIN1

C     COMPUTE DISCRETE SYSTEM MATRICES
      IF (ISFLAG(1) .EQ. 0) GO TO 510
      CALL DSCRT(NX,AS,DEL,AD,AINT,5)
510   IF (ISFLAG(1)+ISFLAG(2) .EQ. 0) GO TO 520
      CALL MMUL(AINT,BS,NX,NX,NU,BD)
520   IF (ISFLAG(8) .EQ. 0) GO TO 540
      DO 530 I=1,NX
      X(I)=X(I)+XINC(I)
530   CONTINUE

C     COMPUTATION OF TIME AVERAGES
540   IF (T .LT. TO+0.0001) GO TO 630
      DO 550 I=1,NX
      XMEAN(I)=XMEAN(I)+X(I)*RNPTS
      XRMS(I)=XRMS(I)+(X(I)**2)*RNPTS
550   CONTINUE
      DO 560 I=1,NY
      YMEAN(I)=YMEAN(I)+Y(I)*RNPTS
      YRMS(I)=YRMS(I)+(Y(I)**2)*RNPTS
560   CONTINUE
      DO 570 I=1,NU
      UMEAN(I)=UMEAN(I)+U(I)*RNPTS
      URMS(I)=URMS(I)+(U(I)**2)*RNPTS
570   CONTINUE

C     COMPUTE XRMS, YRMS URMS AT THE END OF THE RUN
      IF (TIME+DEL+0.0001 .LT. TEND) GO TO 590
      DO 575 I=1,NX
      XRMS(I)=SQRT(XRMS(I)-XMEAN(I)**2)
575   CONTINUE
      DO 580 I=1,NY
      YRMS(I)=SQRT(YRMS(I)-YMEAN(I)**2)
580   CONTINUE
```

```
      DO 585 I=1,NU
      URMS(I)=SQRT(URMS(I)-UMEAN(I)**2)
585   CONTINUE
C     OUTPUT THE MEAN AND RMS VALUES OF THE X, Y AND U VECTORS
      IO=3
      WRITE (KOUT,2000)
2000  FORMAT (17H MEAN OF X VECTOR)
      CALL VECTIO(XMEAN,NX,IO)
      WRITE (KOUT,2010)
2010  FORMAT (23H RMS VALUES OF X VECTOR)
      CALL VECTIO(XRMS,NX,IO)
      WRITE (KOUT,2020)
2020  FORMAT (17H MEAN OF Y VECTOR)
      CALL VECTIO(YMEAN,NY,IO)
      WRITE (KOUT,2030)
2030  FORMAT (23H RMS VALUES OF Y VECTOR)
      CALL VECTIO(YRMS,NY,IO)
      WRITE (KOUT,2040)
2040  FORMAT (17H MEAN OF U VECTOR)
      CALL VECTIO(UMEAN,NU,IO)
      WRITE (KOUT,2050)
2050  FORMAT (23H RMS VALUES OF U VECTOR)
      CALL VECTIO(URMS,NU,IO)
C     INTEGRATE SYSTEM EQUATIONS
590   DO 600 I=1,NX
      DUM(I)=0.0
600   CONTINUE
      IF (NW .GT. 0) CALL VMAT2(DUM,ES,W,NX,NW,DUM)
      IF (NZ .GT. 0) CALL VMAT2(DUM,FS,Z,NX,NZ,DUM)
      DO 605 I=1,NX
      DUM(I)=DUM(I)*DEL
605   CONTINUE
      CALL VMAT2(DUM,AD,X,NX,NX,DUM)
      CALL VMAT2(DUM,BD,U,NX,NU,X)
      CALL VMAT1(CS,X,NY,NX,DUM)
      CALL VMAT2(DUM,DS,U,NY,NU,Y)
C     X(T+DEL) AND Y(T+DEL) HAVE JUST BEEN COMPUTED
C     NOW OBTAIN W AND Z OVER (T,T+DEL) FOR THE NEXT STEP
630   IF (T+0.0001 .GE. TEND) GO TO 650
      IF (NW .EQ. 0) GO TO 640
      DO 635 I=1,NW
      C1=SQRT(W0(I)/DEL)
      W(I)=GAUSS(IXYZ)*C1
635   CONTINUE
640   IF (NZ .EQ. 0) GO TO 650
      DO 645 I=1,NZ
      Z(I)=FDET(I,T)
645   CONTINUE
650   RETURN
      END
```

```
      SUBROUTINE MAN(IDUM,Y,U,Z,XH,YHAT,RES)
C     INTEGRATES MAN EQUATIONS ONE TIME STEP
C     CALLS SUBROUTINES FOR EXTERNAL AND INTERNAL MAN UPDATES

      DIMENSION
     1 Y(30), U(30), Z(30), PASTY(15,11), P(30), XH(30), YHAT(30),
     2 VUO(4), RES(30), IMFLAG(32), AVGY2(30), AVGU(4), AVGU2(4),
     3 VYO(30), FGAIN(15,15), PASTUC(4,11)

      COMMON
     1 /INOU/   KIN, KOUT, KPTR, KPUNCH,
     1          KDISK, IPOS, IGOS
     3 /MAIN1/  NDIM, NDIM1, COM1(1)
     4 /MAIN2/  COM2(1)
     5 /TIMES/  TIME, DEL ,TO, TEND, NPRNT
     E /MANX/   NXM, NUM, BM(15,4), AM(225)
     F /MANAD/  BD(15,4), AD(15,15)
     G /MANY/   NYM, DM(15,4), CM(225)
     H /MANW/   NWM, WOM(8), EM(60)
     I /MANZ/   NZM, FM(60)
     J /MANINC/ TD, NPRED, XHINC(30)
     K /RATIOS/ PU(30), VU(30), PY(30), VY(30), TH(30), ATTN(30),
     K          SIGMA(15,15)
     L /GAINBK/ CGN(225)

C     INITIALIZE VECTORS AND MATRICES IF TIME IS LESS THAN TO
      IF (TIME .GT. TO+0.0001) GO TO 100
      DO 10 I=1,30
      P(I)=0.0
      XH(I)=0.0
      YHAT(I)=0.0
      RES(I)=Y(I)
      AVGY2(I)=Y(I)*Y(I)
10    CONTINUE
      DO 12 I=1,4
      DO 11 J=1,11
      PASTUC(I,J)=0.0
11    CONTINUE
      VUO(I)=0.0
      AVGU(I)=0.0
      AVGU2(I)=0.0
12    CONTINUE
      KOUNT=0
      CALL IDENT(NDIM,SIGMA,1.0E-05)
      TCOR=1.0
      TMEM=0.5
      ALPHA=EXP(-DEL/TMEM)
      TPR=TMEM
      DO 13 I=1,15
      DO 13 J=1,11
      PASTY(I,J)=Y(I)
13    CONTINUE

C     DO EXTERNAL AND INTERNAL MAN UPDATES
100   CALL UPDATE(IMFLAG)
      NTOT=NXM+NUM
      IF (IMFLAG(8) .EQ. 1) CALL VADD(NXM,1.0,P,XHINC)
      TPR=TPR*ALPHA+DEL
      LOC=NPRED+1
      DO 110 I=1,NYM
      PASTY(I,LOC)=Y(I)
      C2=PASTY(I,1)
      AVGY2(I)=ALPHA*AVGY2(I)+C2*C2*DEL
      C1=ABS(C2)
      C1=XGAIN(TH(I),0.0,C1)
      C3=C1*C1*ATTN(I)
```

```
      C1=PY(I)
      IF (KOUNT .LT. NPRED) C1=1.0E+10
      VYO(I)=(C2*C2*C1+VY(I))/C3
      RES(I)=SQRT(VYO(I)/DEL)*GAUSS(IDUM)+C2
      RES(I)=RES(I)-DOT3(NTOT,CM(I),P)
      VYO(I)=(AVGY2(I)*C1/TPR+VY(I))/C3
110   CONTINUE
C     UPDATE THE KALMAN FILTER ESTIMATES
      CALL MMUL(CM,SIGMA,NYM,NTOT,NTOT,COM1)
      CALL MAT2(NYM,NTOT,COM1,CM,FGAIN)
      DO 120 I=1,NYM
      FGAIN(I,I)=FGAIN(I,I)+VYO(I)/DEL
120   CONTINUE
      CALL GMINV(NYM,NYM,FGAIN,COM2,MRANK,0)
      CALL MAT5A(COM1,COM2,NTOT,NYM,NYM,FGAIN)
      CALL VMAT2(P,FGAIN,RES,NTOT,NYM,P)
      CALL MMUL(FGAIN,CM,NTOT,NYM,NTOT,COM2)
      CALL DIAG2(NTOT,COM2,COM2,-1.0,1.0)
      CALL MMUL(COM2,SIGMA,NTOT,NTOT,NTOT,COM1)
      DO 130 I=1,NYM
      C1=VYO(I)/DEL
      DO 130 J=1,NTOT
      SIGMA(J,I)=C1*FGAIN(J,I)
130   CONTINUE
      CALL MAT2(NTOT,NYM,FGAIN,SIGMA,SIGMA)
      CALL MAT6S(NTOT,NTOT,COM1,COM2,SIGMA)
C     OBTAIN PREDICTION OF CURRENT STATE
      DO 140 I=1,NTOT
      YHAT(I)=0.0
      XH(I)=P(I)
140   CONTINUE
      IF (NPRED.EQ.0) GO TO 170
      LOC1=NPRED+1
      DO 150 L=1,NPRED
      CALL VMAT1(AD,XH,NTOT,NTOT,YHAT)
      CALL VMAT2(YHAT,BD,PASTUC(1,L),NTOT,NUM,XH)
150   CONTINUE
170   CALL VMAT1(CGN,XH,NUM,NXM,PASTUC(1,LOC))
      DEL2=-0.5*DEL
      DO 172 I=1,NUM
      C1=ABS(VUO(I))
      PASTUC(I,LOC)=DEL2*PASTUC(I,LOC)
      YHAT(I)=DEL2*(U(I)+SQRT(C1/DEL)*GAUSS(IDUM))
      U(I)=U(I)+PASTUC(I,LOC)
172   WOM(I+NWM)=C1
      II=NDIM*NXM+1
      CALL VMAT2(U,CGN(II),YHAT,NUM,NUM,U)
      DO 175 I=1,NUM
      AVGU(I)=ALPHA*AVGU(I)+U(I)*DEL
      AVGU2(I)=ALPHA*AVGU2(I)+U(I)*U(I)*DEL
      C1=AVGU(I)/TPR
      VUO(I)=(AVGU2(I)/TPR-C1*C1)*PU(I)+VU(I)
175   CONTINUE
C     PROPAGATE SIGMA, P
      CALL VMAT1(AD,P,NTOT,NTOT,YHAT)
      CALL VMAT2(YHAT,BD,PASTUC,NTOT,NUM,P)
      CALL VMAT1(CM,XH,NYM,NTOT,YHAT)
      CALL MMUL(AD,SIGMA,NTOT,NTOT,NTOT,COM1)
      NWU=NWM+NUM
      II=1
      DO 200 I=1,NWU
      C1=WOM(I)*DEL
      CALL VSCALE(COM2(II),EM(II),NTOT,C1)
```

```
      II=II+NDIM
200   CONTINUE
      CALL MAT2(NTOT,NWU,COM2,EM,SIGMA)
      IF (NZM.EQ.0) GO TO 220
      II=1
      DO 210 I=1,NZM
      C1=Z(I)*Z(I)*TCOR*DEL
      CALL VSCALE(COM2(II),FM(II),NXM,C1)
      II=II+NDIM
210   CONTINUE
      CALL MAT6S(NXM,NZM,COM2,FM,SIGMA)
220   CALL MAT6S(NTOT,NTOT,COM1,AD,SIGMA)
      IF (NPRED.EQ.0) GO TO 235
      CALL EQUATE(PASTY,PASTY(1,2),NYM,NPRED)
      DO 230 I=1,NPRED
      DO 230 J=1,NUM
      PASTUC(J,I)=PASTUC(J,I+1)
230   CONTINUE
235   KOUNT=KOUNT+1
      IF (KOUNT.GT.NPRED) KOUNT=NPRED
      RETURN

C     DUMMY CALL TO MAT6 TO FORCE LOADING
      CALL MAT6(NTOT,NTOT,COM1,COM1,COM1)

      END
```

```
      FUNCTION GAUSS(DUM)
C     RETURNS A GAUSSIAN RANDOM VARIABLE
C     WITH ZERO MEAN AND UNIT STD DEVIATION
C     BY SUMMING 12 UNIFORMLY DISTRIBUTED VARIABLES

      A=0.0
      DO 50 I=1,12
      A=A+RANF(DUM)
50    CONTINUE
      GAUSS=A-6.0
      RETURN

      END
```

```
C       MCIO - I/O FOR MCARLO
C       INCLUDES
C         1 - SUBROUTINE UPDATE
C         2 - SUBROUTINE INFORM
C         3 - SUBROUTINE PUTOUT
C         4 - SUBROUTINE PRINTR

        SUBROUTINE UPDATE(IMFLAG)
C       PERFORM EXTERNAL UPDATES TO THE MAN

        DIMENSION
     1  IMFLAG(20), MCODE(20), NSTEP(32)

        COMMON
     1  /INOU/   KIN, KOUT, KPTR, KPUNCH,
     1           KDISK
     3  /MAIN1/  NDIM, NDIM1, COM1(1)
     5  /TIMES/  TIME, DEL, TO, TEND, NPRNT
     8  /SYSX/   NX, NU, B(15,4), A(1)
     A  /SYSY/   NY, D(15,4), C(1)
     B  /SYSW/   NW, WO(4), E(1)
     C  /SYSZ/   NZ, F(1)
     E  /MANX/   NXM, NUM, BM(15,4), AM(15,15)
     F  /MANAD/  BD(15,4), AD(15,15)
     G  /MANY/   NYM, DM(15,4), CM(15,15)
     H  /MANW/   NWM, WOM(8), EM(15,4)
     I  /MANZ/   NZM, FM(15,4)
     J  /MANINC/ TD, NPRED, XHINC(30)
     K  /RATIOS/ PU(30), VU(30), PY(30), VY(30), TH(30), ATTN(30),
     K           SIGMA(15,15)
     L  /GAINBK/ CGN(15,15)

        DATA
     1  NMCODE, LEND, PI, LTIME
     1  /20, 4HENDM, 3.14159, 4HTIME/,
     2  MCODE
     2  /2HAM, 2HBM, 2HCM, 2HDM, 2HEM, 2HFM, 3HWOM, 5HXHINC,
     2   2HTD, 3HMNA, 3HMNR, 3HSNA, 3HSNR, 2HTH, 4HATTN,
     2   5HCGAIN, 5HDGAIN, 3HINT, 5HPRINT, 5HDUMMY/

        IF (TIME .GT. TO+0.0001) GO TO 100

C       INITIALIZATION
C       SPECIFY THE MAN INTERNAL BREAKS
        READ (KIN,1050) (NSTEP(I), I=1,NMCODE)
 1050   FORMAT (16I5)
        WRITE (KOUT,1060)
 1060   FORMAT (23H HUMAN INTERNAL BREAKS ,17H INDEX  CODE  NDT)
        DO 65 I=1,NMCODE
        IF (NSTEP(I) .LE. 0) GO TO 65
        WRITE (KOUT,1070) I, MCODE(I), NSTEP(I)
 1070   FORMAT (27X,I2,3X,A5,1X,I5)
 65     CONTINUE

C       PARAMETER INITIALIZATION
        TNEXT=TO
        DO 80 I=1,30
        PU(I)=0.0
        XHINC(I)=0.0
        VU(I)=0.0
        PY(I)=0.0
        VY(I)=0.0
        TH(I)=0.0
        ATTN(I)=1.0
 80     CONTINUE
```

```
      NPRED=0
      DO 85 I=1,NDIM
      DO 85 J=1,4
      DM(I,J)=0.0
85    CONTINUE
      TD=0.0
      NWM=0
      NZM=0

C     TAKE CARE OF INTERNAL MAN BREAKS
100   DO 105 I=1,NMCODE
      IMFLAG(I)=0
      IF (NSTEP(I) .EQ. 0) GO TO 105
      ITME=IFIX((TIME-TO+0.0001)/DEL)
      IF (MOD(ITME,NSTEP(I)) .EQ. 0) CALL MANNEW(I)
105   CONTINUE

C     TAKE CARE OF EXTERNAL MAN BREAKS
110   IF (TIME+0.0001 .LT. TNEXT) GO TO 500
120   READ (KIN,1130) IDEN, NQQ, BRKT
1130  FORMAT (A5,2X,I3,E10.0)
      IF (IDEN.NE.LEND) GO TO 140
135   TNEXT=1.0E+05
      GO TO 110
140   IF (IDEN.NE.LTIME) GO TO 150
      TNEXT=BRKT
      GO TO 110

C     SEARCH THROUGH THE UPDATE CODES
150   DO 160 KEY=1,NMCODE
      IF (IDEN.EQ.MCODE(KEY)) GO TO 170
160   CONTINUE

C     CODE WAS ILLEGAL
      WRITE (KOUT,1165) IDEN
1165  FORMAT (23H ILLEGAL INPUT CODE OF ,A5)
      CALL EXIT

C     DO THE SPECIFIED MAN EXTERNAL UPDATE
170   IMFLAG(KEY)=1
      WRITE (KOUT,1175) TIME, IDEN, NQQ
1175  FORMAT (/,27H HUMAN EXTERNAL BREAK AT T= ,F8.3,4X,4HCODE,2X,A5,
     1 4X,7HINDEX= ,I3)
      IF (NQQ.LE.0 .AND. KEY.LE.7) WRITE (KOUT,1180)
1180  FORMAT (21H HUMAN MODEL = SYSTEM)
      IO=2
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19), KEY

C     SYSTEM DYNAMICS - AM, BM, CM, DM, EM
1     NXM=NQQ
      IF (NXM.LE.0) GO TO 201
      CALL MATIO(AM,NXM,NXM,IO)
      GO TO 120
201   NXM=NX
      CALL EQUATE(AM,A,NXM,NXM)
      GO TO 120
2     NUM=NQQ
      IF (NUM.LE.0) GO TO 202
      CALL MATIO(BM,NXM,NUM,IO)
      GO TO 120
202   NUM=NU
      CALL EQUATE(BM,B,NXM,NUM)
      GO TO 120
3     NYM=NQQ
      IF (NYM.LE.0) GO TO 203
      CALL MATIO(CM,NYM,NXM,IO)
```

```
      GO TO 120
203   NYM=NY
      CALL EQUATE(CM,C,NYM,NXM)
      GO TO 120
4     NYM=NQQ
      IF (NYM.LE.0) GO TO 204
      CALL MATIO(DM,NYM,NUM,IO)
      GO TO 120
204   NYM=NY
      CALL EQUATE(DM,D,NYM,NXM)
      GO TO 120
5     NWM=NQQ
      IF (NWM.LE.0) GO TO 205
      CALL MATIO(EM,NXM,NWM,IO)
      GO TO 120
205   NWM=NW
      IF (NW.GT.0) CALL EQUATE(EM,E,NXM,NWM)
      GO TO 120

C     DETERMINISTIC INPUT (FM MATRIX) - FM
6     NZM=NQQ
      IF (NZM.LE.0) GO TO 206
      CALL MATIO(FM,NXM,NZM,IO)
      GO TO 120
206   NZM=NZ
      IF (NZ.GT.0) CALL EQUATE(FM,F,NXM,NZM)
      GO TO 120

C     DRIVING NOISE - WOM
7     NWM=NQQ
      IF (NWM.LE.0) GO TO 207
      CALL VECTIO(WOM,NWM,IO)
      GO TO 120
207   NWM=NW
      IF (NW.GT.0) CALL EQUATE(WOM,WO,NWM,1)
      GO TO 120

C     INCREMENT TO STATE ESTIMATES - XHINC
8     CALL VECTIO(XHINC,NXM,IO)
      GO TO 120

C     TIME DELAY - TD
9     CALL VECTIO(TD,1,IO)
      NPRED=IFIX(TD/DEL+0.5001)
      TD=DEL*NPRED
      GO TO 120

C     NOISES - MNA, MNR, SNA, SNR
10    CALL VECTIO(VU,NUM,IO)
      GO TO 120
11    CALL VECTIO(PU,NUM,IO)
      DO 211 I=1,NUM
      PU(I)=PI*10.0**(PU(I)/10.0)
211   CONTINUE
      GO TO 120
12    CALL VECTIO(VY,NYM,IO)
      GO TO 120
13    CALL VECTIO(PY,NYM,IO)
      DO 213 I=1,NYM
      PY(I)=PI*10.0**(PY(I)/10.0)
213   CONTINUE
      GO TO 120

C     THRESHOLDS AND ATTENTION - TH, ATTN
14    CALL VECTIO(TH,NYM;IO)
      GO TO 120
```

```
15        CALL VECTIO(ATTN,NYM,IO)
          GO TO 120

C         CONTINUOUS AND DISCRETE CONTROL GAINS - CGAIN, DGAIN
16        CALL MATIO(CGN,NUM,NUM+NXM,IO)
          GO TO 120
17        CALL MATIO(CGN,NUM,NUM+NXM,IO)
          GO TO 120

C         CALL AN INTERNAL MAN UPDATE - INT
18        CALL MANNEW(NQQ)
          GO TO 120

C         PRINT INTERVAL - PRINT
19        NPRNT=NQQ
          GO TO 120

C         NO MORE MAN UPDATES AT THIS TIME
500       CONTINUE
C         UPDATE VARIOUS MAN PARAMETERS
          NTOT=NXM+NUM
C         COMPUTE DISCRETE MAN MATRICES
          IF (IMFLAG(1)+IMFLAG(2) .EQ. 0) GO TO 520
          DO 510 I=1,NUM
          II=NXM+I
          DO 505 J=1,NXM
          AD(II,J)=0.0
505       CONTINUE
          DO 506 K=1,NUM
          BD(II,K)=0.0
506       CONTINUE
          BD(II,I)=1.0
510       CONTINUE
          CALL DSCRT(NXM,AM,DEL,AD,COM1,5)
          CALL MMUL(COM1,BM,NXM,NXM,NUM,BD)
          CALL EQUATE(AM(1,NXM+1),BM,NXM,NUM)

C         INCORPORATE NEW CGAINS
520       IF (IMFLAG(16).EQ.0) GO TO 540
          CALL MSCALE(AM(NXM+1,1),CGN,NUM,NTOT,-1.0)
          CALL DSCRT(NTOT,AM,DEL,CGN,COM1,5)
          CALL MMUL(AM(NXM+1,1),COM1,NUM,NTOT,NTOT,CGN)
          DO 525 I=1,NUM
          II=I+NXM
          DO 525 J=1,NTOT
          AM(II,J)=0.0
          CGN(I,J)=-CGN(I,J)/DEL
525       CONTINUE
          WRITE (KOUT,5250)
5250      FORMAT (37H EQUIVALENT DISCRETE GAINS GENERATED )
          IO=3
          CALL MATIO(CGN,NUM,NTOT,IO)

C         UPDATE EM AND CM
540       CALL MMUL(BD,CGN(1,NXM+1),NTOT,NUM,NUM,EM(1,NWM+1))
          CALL EQUATE(CM(1,NXM+1),DM,NYM,NUM)
          DO 550 J=1,NUM
          JJ=J+NXM
          JQ=J+NWM
          DO 550 I=1,NTOT
          EM(I,JQ)=0.5*EM(I,JQ)
          AD(I,JJ)=BD(I,J)-DEL*EM(I,JQ)
550       CONTINUE
          IF (IMFLAG(5)+IMFLAG(7).EQ.0) GO TO 570
          IQ=NXM+1
          IF (NWM.LE.0) GO TO 570
```

```
      DO 560 J=1,NWM
      DO 560 I=IQ,NTOT
560   EM(I,J)=0.0
570   CONTINUE
      RETURN

      END
```

```
      SUBROUTINE INFORM(ISTEP,X,Y,U,XH,YH,RES)
C     DO OUTPUT FOR A SINGLE TIME STEP

      DIMENSION
     1 X(1), Y(1), U(1), XH(1), YH(1), RES(1), DUM1(21), DUM2(21)

      COMMON
     1 /INOU/   KIN, KOUT, KPTR, KPUNCH,
     1          KDISK, IPOS, IGOS
     5 /TIMES/  TIME, DEL, TO, TEND, NPRNT
     6 /INFO/   NREC, NPRINT, NPLOT, LPP(20,6), IP(6), IG(6),
     6          SMIN(21), SMAX(21)

C     CHECK IF TIME FOR SOME OUTPUT
      IF (MOD(ISTEP,NPRNT) .NE. 0) RETURN

C     DO SOME INITIALIZATION
      IF (TIME .GT. TO+0.0001) GO TO 100
      NPRINT=1
      NPLOT=1
      REWIND IPOS
      REWIND IGOS
      DO 8 I=1,21
      SMIN(I)=1.0E+20
      SMAX(I)=-1.0E+20
8     CONTINUE
      DO 10 I=1,6
      IP(I)=0
      IG(I)=0
      DO 9 J=1,20
      L1=LPP(J,I)
      IF (L1.EQ.1 .OR. L1.EQ.2) IP(I)=IP(I)+1
      IF (L1.EQ.2 .OR. L1.EQ.3) IG(I)=IG(I)+1
9     CONTINUE
      NPRINT=NPRINT+IP(I)
      NPLOT=NPLOT+IG(I)
10    CONTINUE
      NREC=0
      IF (ISTEP.EQ.0) RETURN

C     DO OUTPUT FOR THE CURRENT TIME
100   DUM1(1)=TIME
      DUM2(1)=TIME
      IF (SMIN(1).GE.TIME) SMIN(1)=TIME
      SMAX(1)=TIME
      LOC1=1
      LOC2=1
      DO 160 J=1,6
      DO 160 I=1,20
      L1=LPP(I,J)
      IF (L1.GE.3) GO TO 140
      IF (L1.LE.0) GO TO 160
      LOC1=LOC1+1
      IF (J.EQ.1) DUM1(LOC1)=X(I)
      IF (J.EQ.2) DUM1(LOC1)=Y(I)
      IF (J.EQ.3) DUM1(LOC1)=U(I)
      IF (J.EQ.4) DUM1(LOC1)=XH(I)
      IF (J.EQ.5) DUM1(LOC1)=YH(I)
      IF (J.EQ.6) DUM1(LOC1)=RES(I)
140   IF (L1.EQ.1) GO TO 160
      LOC2=LOC2+1
      IF (J.EQ.1) DUM2(LOC2)=X(I)
      IF (J.EQ.2) DUM2(LOC2)=Y(I)
      IF (J.EQ.3) DUM2(LOC2)=U(I)
      IF (J.EQ.4) DUM2(LOC2)=XH(I)
      IF (J.EQ.5) DUM2(LOC2)=YH(I)
```

```
      IF (J.EQ.6) DUM2(LOC2)=RES(I)
150   C1=DUM2(LOC2)
      IF (SMIN(LOC2).GE.C1) SMIN(LOC2)=C1
      IF (SMAX(LOC2).LE.C1) SMAX(LOC2)=C1
160   CONTINUE
      IF (LOC1.GT.1) CALL PUTOUT(DUM1,NPRINT,IPOS)
      IF (LOC2.GT.1) CALL PUTOUT(DUM2,NPLOT,IGOS)
      NREC=NREC+1
      RETURN

      END
```

```
      SUBROUTINE PUTOUT(DUM,NVAR,IDISK)
C     SUBROUTINE TO SAVE OUTPUT ON A FILE

      DIMENSION
     1 DUM(1)

      WRITE (IDISK) (DUM(I), I=1,NVAR)
      RETURN

      END
```

```
      SUBROUTINE PRINTR(NPRINT,NPLOT,DUM1,DUM2)
C     PRINT THE OUTPUT AT THE END OF A RUN

      DIMENSION
     1 DUM1(1), DUM2(1), GRAPH(1350), TITLE(6), LET(11)

      COMMON
     1 /INOU/  KIN, KOUT, KPTR, KPUNCH,
     1         KDISK, IPOS, IGOS
     2 /PLOT1/ NV, NH, NCPW, LW, XL, XH, YL, YH, NXES, NDIR, IST,
     2         NGLV, NGLH, BSYM, GSYM, PSYM, ND1, ND2, NOUT
     3 /MAIN1/ NDIM, NDIM1, STORE(1)
     4 /MAIN2/ COM2(1)
     6 /INFO/  NREC, I1, I2, LPP(20,6), IP(6), IG(6),
     6         SMIN(21), SMAX(21)

      DATA
     1 TITLE
     1 /8H  STATE , 8H OUTPUT , 8HCONTROL , 8H  XMHAT ,
     1  8H  YMHAT , 8H INOVAT /

      IF (NPRINT.EQ.1) GO TO 51
      REWIND IPOS
      DO 10 I=1,NREC
      READ (IPOS) (DUM1(KK), KK=1,NPRINT)
      II=I
      DO 9 L=1,NPRINT
      STORE(II)=DUM1(L)
9     II=II+NREC
10    CONTINUE

      IBEG=NREC+1
      DO 50 I=1,6
      M=0
      DO 30 L=1,20
      IQ=LPP(L,I)
      IF (IQ.EQ.0 .OR. IQ.EQ.3) GO TO 30
      M=M+1
      LET(M)=L
30    CONTINUE
      IF (M.EQ.0) GO TO 50
      CALL PAGEFD(KOUT,1)
      WRITE (KOUT,1035) (TITLE(I), LET(J),J=1,M)
1035  FORMAT (1H ,3X,4HTIME,2X,10(A8,I2,2X))
      LIM1=IBEG
      LIM2=IBEG+(M-1)*NREC
      DO 40 L=1,NREC
      WRITE (KOUT,1045) STORE(L), (STORE(J), J=LIM1,LIM2,NREC)
1045  FORMAT (1H ,F7.2,10(2X,1PE10.3))
      LIM1=LIM1+1
      LIM2=LIM2+1
40    CONTINUE
50    IBEG=IBEG+M*NREC

51    IF (NPLOT.EQ.1) RETURN
      REWIND IGOS

      DO 60 I=1,NREC
      READ (IGOS) (DUM2(KK),KK=1,NPLOT)
      II=I
      DO 59 L=1,NPLOT
      STORE(II)=DUM2(L)
59    II=II+NREC
60    CONTINUE
```

```
      ND2=NREC
      XH=SMAX(1)
      XL=SMIN(1)
      M=1
      DO 70 I=1,6
      DO 70 L=1,20
      IQ=LPP(L,I)
      IF (IQ.LT.2) GO TO 70
      M=M+1
      YH=SMAX(M)
      YL=SMIN(M)
      IBEG=(M-1)*NREC+1
      CALL KPLOT(GRAPH,STORE,STORE(IBEG),0,0,0,0,0)
      WRITE (KOUT,1082) TITLE(I), L
1082  FORMAT (/,1H ,A8,I3)
70    CONTINUE
      RETURN

      END
```

```
      SUBROUTINE KPLOT(W,X,Y,NTAPE,IX,IY,NVAR,Y1)

      COMMON /PLOT1/
     1 NV,NH,NCPW,LW,VLH(4),NXES,NDIR,IST,NGLV,NGLH,BSYM,GSYM,
     2 PSYM,NDIM1,NDIM2,NO

      DIMENSION W(1),X(1),Y(1),Y1(NVAR),STORE(70),Q(4),IPX(4),K(3)

      EQUIVALENCE (Q(1),XL1),(Q(2),XH1),(Q(3),YL1),(Q(4),YH1)
      EQUIVALENCE (ISC,K(1)),(JSC,K(3))

      DATA IPX/3,4,1,2/

      IF (NH.GT.121) NH=121
C     NCPW IS THE NUMBER OF CHARACTERS PER WORD
C     (60 BIT WORD 6 BIT DISPLAY CODE ON CDC)
      NCPW=10
      LW=NH/NCPW+1
      IF ((IST/10).GT.0) NCOUNT=0
      NCOUNT=NCOUNT+1
      IF (NCOUNT.EQ.10) IST=1
      L=1
      DO 10 I=1,4
      Q(I)=-1.0E08*(-1)**I
      K(L)=1
      IF (VLH(L).EQ.VLH(L+1)) GO TO 10
      K(L)=0
      Q(I)=VLH(I)
10    IF (I.EQ.2) L=3
      IF (NTAPE.EQ.0) GO TO 1200

C     SKIP THIS PART IF PLOTTING FROM CORE
      IFLAG=0
      GO TO 40
1600  IFLAG=1
40    NN=0
      REWIND NTAPE
50    READ (NTAPE) Y1
C     GO TO 2800 ON EOF
      IF (EOF(NTAPE))2800,100
100   NN=NN+1
      IF (NN.LT.NDIM1) GO TO 50
      IF (IFLAG.EQ.1) GO TO 1700
      IF (ISC+JSC.EQ.0) GO TO 1710
300   NN=NN+1
      IF (ISC.EQ.0) GO TO 600
      XL1=AMIN1(XL1,Y1(IX))
      XH1=AMAX1(XH1,Y1(IX))
      IF (JSC.EQ.0) GO TO 200
600   YL1=AMIN1(YL1,Y1(IY))
      YH1=AMAX1(YH1,Y1(IY))
200   READ (NTAPE) Y1
      IF (EOF(NTAPE))1600,210
210   IF (NN-NDIM2) 300,300,1600

C     RESUME HERE
1200  IF (ISC.EQ.0) GO TO 1400
      DO 1300 I=NDIM1,NDIM2
      XL1=AMIN1(XL1,X(I))
1300  XH1=AMAX1(XH1,X(I))
1400  IF (JSC.EQ.0) GO TO 1700
      DO 1500 I=NDIM1,NDIM2
      YL1=AMIN1(YL1,Y(I))
1500  YH1=AMAX1(YH1,Y(I))
1700  IF (ISC.EQ.1) CALL ADJUST(XH1,XL1)
      IF (JSC.EQ.1) CALL ADJUST(YH1,YL1)
```

```
1710    IF (NDIR/10) 1720,1740,1720
1720    TMP=XL1
        XL1=XH1
        XH1=TMP
1740    IF (NDIR-10*(NDIR/10)) 1760,1780,1760
1760    TMP=YL1
        YL1=YH1
        YH1=TMP
1780    J=7*(NCOUNT-1)+1
        IF (J.EQ.1) CALL QINIT(W)
        STORE(J)=PSYM
        DO 1800 I=1,4
        IF (NXES.EQ.0) L=I+J
        IF (NXES.GT.0) L=IPX(I)+J
1800    STORE(L)=Q(I)
        STORE(J+5)=(NH-1)/(STORE(J+2)-STORE(J+1))
        STORE(J+6)=(NV-1)/(STORE(J+4)-STORE(J+3))
2200    IF (NTAPE.EQ.0) GO TO 2500

C       SKIP THIS PART IF PLOTTING FROM CORE
        DO 2400 I=NDIM1,NDIM2
        IF (NXES.EQ.0) CALL KPLOTC(STORE(J),W,Y1(IX),Y1(IY))
        IF (NXES.GT.0) CALL KPLOTC(STORE(J),W,Y1(IY),Y1(IX))
2400    READ (NTAPE) Y1
        IF (EOF(NTAPE)) 2800,2700

C       SKIP THIS PART IF PLOTTING FROM A FILE
2500    DO 2600 I=NDIM1,NDIM2
        IF (NXES.EQ.0) CALL KPLOTC(STORE(J),W,X(I),Y(I))
2600    IF (NXES.GT.0) CALL KPLOTC(STORE(J),W,Y(I),X(I))

C       RESUME HERE
2700    IF ((IST-10*(IST/10)).GT.0) CALL QPRINT(W,NO,NCOUNT,STORE)
        RETURN

C       ERROR MESSAGE
2800    WRITE (NO,2900)
2900    FORMAT(//32H INSUFFICIENT DATA ON INPUT FILE,/,
     1  1H ,28H PLOTTING ROUTINE TERMINATED)
        RETURN
        END
```

```
SUBROUTINE ADJUST(XH1,XL1)
IF (XH1.EQ.XL1) XL1=0.9*XL1-10.0
A=IFIX(100.0+ALOG10(XH1-XL1))-100.0
XH1T=XH1*10.0**(1.0-A)
XL1T=XL1*10.0**(1.0-A)
IF (XH1T .GE. 0.0) XH1T=IFIX(XH1T+0.9)
XH1T=IFIX(XH1T)
IF (XL1T .LE. 0.0) XL1T=IFIX(XL1T-0.9)
XL1T=IFIX(XL1T)
XH1=XH1T*10.0**(A-1.0)
XL1=XL1T*10.0**(A-1.0)
RETURN
END
```

```
      SUBROUTINE QINIT(IMAGE)

      COMMON /PLOT1/
     1 NV,NH,NCPW,LW,Q(4),NXES,NDIR,IST,NGLV,NGLH,BSYM,GSYM

      DIMENSION IMAGE(1)

      DATA IBLNK/10H

      N=LW*NV
      DO 100 I=1,N
100   IMAGE(I)=IBLNK
      DO 101 I=1,NH
      CALL QPLOT(IMAGE,I,1,BSYM)
101   CALL QPLOT(IMAGE,I,NV,BSYM)
      DO 102 I=1,NV
      CALL QPLOT(IMAGE,1,I,BSYM)
102   CALL QPLOT(IMAGE,NH,I,BSYM)
1800  IF (NGLV.EQ.0) GO TO 2000
      NGLV1=NGLV+1
      NH1=NH-1
      DO 1900 I=NGLV1,NH1,NGLV
      DO 1900 J=1,NV
1900  CALL QPLOT(IMAGE,I,J,GSYM)
2000  IF (NGLH.EQ.0) RETURN
      NGLH1=NGLH+1
      NV1=NV-1
      DO 2100 I=NGLH1,NV1,NGLH
      DO 2100 J=1,NH
2100  CALL QPLOT(IMAGE,J,I,GSYM)
      RETURN
      END
```

```
      SUBROUTINE KPLOTC(W,IMAGE,X,Y)

      DIMENSION W(1), IMAGE(1)
      COMMON /PLOT1/ NV, NH

      J=(X-W(2))*W(6)+1.5
      IF ((J.LE.0).OR.(J.GT.NH)) RETURN
      I=NV-IFIX((Y-W(4))*W(7)+0.5)
      IF ((I.LE.0).OR.(I.GT.NV)) RETURN
      CALL QPLOT (IMAGE,J,I,W(1))
      RETURN
      END
```

```
      SUBROUTINE QPLOT(IMAGE,J,I,SYM)

      COMMON /PLOT1/ NV, NH, NCPW, LW
      DIMENSION IMAGE(1)

      II=J/NCPW
      L=J-NCPW*II
      II=II+1
      IF (L) 101,101,102
101   L=NCPW
      II=II-1
102   IW=II+(I-1)*LW
      CALL PLACE(IMAGE(IW),L,SYM,1)
103   RETURN
      END
```

```
      SUBROUTINE PLACE(A,N,B,M)
C     THE MTH CHAR OF B REPLACES
C     THE NTH CHARACTER OF A
C     CHAR POSITIONS ARE 1 TO 10 FROM LEFT TO RIGHT

      COMMON/INOU/KIN,KOUT
      INTEGER A, B, BX, BY
      DATA MASK/77B/

C     CHECK FOR VALID ARGUMENTS
      IF (N.GT.10 .OR. M.GT.10) GO TO 900
      IF (N.LT.1 .OR. M.LT.1) GO TO 900

C     NULL ALL BUT THE MTH CHAR OF B, PUT IT IN BX
C     NULL THE NTH CHAR OF A
      NSHFT=60-6*N
      MSHFT=60-6*M
      MASKBY = SHIFT(MASK,NSHFT)
      MASKB = SHIFT(MASK,MSHFT)
      MASKA = COMPL(MASKBY)
      A = AND(A,MASKA)
      BX = AND(B,MASKB)

C     SHIFT THE MTH CHAR OF BX TO THE NTH POSITION
C     PUT IT IN BY AND NULL ALL BUT THE NTH CHAR
      MNSHFT=6*(M-N)
      BY = SHIFT(BX,MNSHFT)
      BY = AND(BY,MASKBY)

C     COMBINE A AND BY
      A = OR(A,BY)

      RETURN

C     N OR M OUT OF BOUNDS
900   WRITE (KOUT,1900)
1900  FORMAT (1H ,20HERROR IN SUBR. PLACE,/,
     1 24H N OR M IS OUT OF BOUNDS)
      CALL EXIT

      END
```

```
      SUBROUTINE QPRINT(IMAGE,NO,NCOUNT,STORE)

      DIMENSION IMAGE(1), STORE(1)
      COMMON /PLOT1/ NV, NH, NCPW, LW

      CALL PAGEFD(NO,1)
      DO 110 I=1,NCOUNT
      IB=7*(I-1)+1
110   WRITE (NO,102) STORE(IB+1),STORE(IB),STORE(IB),STORE(IB+2)
      NCANT=NV-NCOUNT
      IA=1
      DO 150 I=1,NV
      IB=I*LW
      IF (I.GT.NCOUNT) GO TO 120
      IBASE=(I-1)*7+1
      WRITE (NO,103) STORE(IBASE),STORE(IBASE+4),(IMAGE(J),J=IA,IB)
      GO TO 150
120   IF (I.GT.NCANT) GO TO 130
      WRITE (NO,105) (IMAGE(J),J=IA,IB)
      GO TO 150
130   IBASE=(I-1-NCANT)*7+1
      WRITE (NO,103) STORE(IBASE),STORE(IBASE+3),(IMAGE(J),J=IA,IB)
150   IA=IA+LW
102   FORMAT(1H ,11X,1PE10.3,1X,A1,77X,A1,1PE10.3)
103   FORMAT(1H ,A1,1PE9.2,1X,12A10,A1)
105   FORMAT(1H ,11X,12A10,A1)
      RETURN
      END
```

```
      SUBROUTINE DSCRT(N,A,DEL,EA,EAINT,NT)
      DIMENSION A(1),EA(1),EAINT(1),COEF(30)
C        SETS EA=EXP(A*DEL),EAINT=INTEGRAL EA 0 TO DEL
      COMMON/MAIN1/NDIM,NDIM1
      NN=N*NDIM
      NTM1=NT-1
      COEF(NT)=1.
      DO 10 I=1,NTM1
      II=NT-I
10    COEF(II)=DEL*COEF(II+1)/FLOAT(I)
C                 NT MUST BE AT LEAST 3
      II=1
      DO 30 I=1,N
      DO 20 J=I,NN,NDIM
20    EAINT(J)=A(J)*COEF(1)
      EAINT(II)=EAINT(II)+COEF(2)
30    II=II+NDIM1
      DO 60 L=3,NT
      T1=COEF(L)
      CALL MMUL(A,EAINT,N,N,N,EA)
      IF(L.EQ.NT)GO TO 70
      II=1
      DO 60 I=1,N
      DO 50 J=I,NN,NDIM
50    EAINT(J)=EA(J)
      EAINT(II)=EAINT(II)+T1
60    II=II+NDIM1
70    DO 80 II=1,NN,NDIM1
      EA(II)=EA(II)+T1
80    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GMINV(NR,NC,A,U,MR,MT)
      DIMENSION A(1),U(1),S(30)
      COMMON/MAIN1/ NDIM,NDIM1
      COMMON/INOU/ KIN,KOUT
      TOL=1.E-12
      MR=NC
      NRM1=NR-1
      TOL1=1.E-20
      JJ=1
      DO 100 J=1,NC
      FAC=DOT(NR,A(JJ),A(JJ))
      JM1=J-1
      JRM=JJ+NRM1
      JCM=JJ+JM1
      DO 20 I=JJ,JCM
20    U(I)=0.
      U(JCM)=1.0
      IF(J.EQ.1) GO TO 54
      KK=1
      DO 30 K=1,JM1
      IF(S(K).EQ.1.0) GO TO 30
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
30    KK=KK+NDIM
      DO 50 L=1,2
      KK=1
      DO 50 K=1,JM1
      IF(S(K).EQ.0.) GO TO 50
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(NR,TEMP,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
50    KK=KK+NDIM
      TOL1=TOL*FAC
      FAC=DOT(NR,A(JJ),A(JJ))
54    IF(FAC.GT.TOL1) GO TO 70
      DO 55 I=JJ,JRM
55    A(I)=0.
      S(J)=0.
      KK=1
      DO 65 K=1,JM1
      IF(S(K).EQ.0.) GO TO 65
      TEMP=-DOT(K,U(KK),U(JJ))
      CALL VADD(NR,TEMP,A(JJ),A(KK))
65    KK=KK+NDIM
      FAC=DOT(J,U(JJ),U(JJ))
      MR=MR-1
      GO TO 75
70    S(J)=1.0
      KK=1
      DO 72 K=1,JM1
      IF(S(K).EQ.1.) GO TO 72
      TEMP=-DOT(NR,A(JJ),A(KK))
      CALL VADD(K,TEMP,U(JJ),U(KK))
72    KK=KK+NDIM
75    FAC=1./SQRT(FAC)
      DO 80 I=JJ,JRM
80    A(I)=A(I)*FAC
      DO 85 I=JJ,JCM
85    U(I)=U(I)*FAC
100   JJ=JJ+NDIM
      IF(MR.EQ.NR.OR.MR.EQ.NC) GO TO 120
      IF(MT.NE.0) WRITE (KOUT,110) NR,NC,MR
110   FORMAT(I3,1HX,I2,8H M  RANK,I2)

120   NEND=NC*NDIM
      JJ=1
```

```
      DO 135 J=1,NC
      DO 125 I=1,NR
      II=I-J
      S(I)=0.
      DO 125 KK=JJ,NEND,NDIM
125   S(I)=S(I)+A(II+KK)*U(KK)
      II=J
      DO 130 I=1,NR
      U(II)=S(I)
130   II=II+NDIM
135   JJ=JJ+NDIM1
      RETURN
      END
```

```
      SUBROUTINE MAT2(N1,N2,X,Y,Z)
C     Z=XY' X,Y=N1*N2,Z=Z'
C     Z AND Y CAN BE EQUIVALENT
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM,NDIM1
      NN2=N2*NDIM
      II=1
      DO 10 I=1,N1
      IJ=II
      DO 5 J=I,N1
      Z(IJ)=DOT2(NN2,X(I),Y(J))
5     IJ=IJ+NDIM
      J=II
      IJ=J
3     IJ=IJ-NDIM
      IF(IJ.LT.I) GO TO 10
      J=J-1
      Z(IJ)=Z(J)
      GO TO 3
10    II=II+NDIM1
      RETURN
      END
```

```
      SUBROUTINE MAT5A(X,Y,N1,N2,N3,Z)
C        Z=XT*Y   X=N2*N1, Y=N2*N3
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      N1M1=N1-1
      NN3=N3*NDIM
      DO 1 I=1,NN3,NDIM
      II=I+N1M1
      DO 1 J=I,II
1     Z(J)=0.0
      ENTRY MAT5AS
      NN3=N3*NDIM
      DO 10 K=1,N2
      KK=K
      DO 8 I=1,N1
      C1=X(KK)
      IF(C1.NE.0.0) CALL VADD1(NN3,C1,Z(I),Y(K))
8     KK=KK+NDIM
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MAT6(N1,N2,X,Y,Z)
C     Z=X*Y´,  WHERE X=N1*N2, Y=N1*N2, Z=Z´=N1*N1
      DIMENSION X(1), Y(1), Z(1)
      COMMON /MAIN1/ NDIM, NDIM1
      NN1=N1*NDIM
      DO 1 I=1,N1
      DO 1 J=I,NN1,NDIM
      Z(J)=0.0
1     CONTINUE
      ENTRY MAT6S
C     Z=Z+X*Y´
      NN2=N2*NDIM
      NN1=N1*NDIM
      DO 6 K=1,NN2,NDIM
      KK=K-1
      J=1
      DO 6 I=1,N1
      C1=Y(I+KK)
      IF (C1.NE.0.0) CALL VADD(I,C1,Z(J),X(K))
      J=J+NDIM
6     CONTINUE
      IF (N1.EQ.1) RETURN
      NN2=NDIM1+1
      DO 10 K=NN2,NN1,NDIM1
      I=K
      J=K
8     I=I-1
      J=J-NDIM
      Z(J)=Z(I)
      IF (J.GT.NDIM) GO TO 8
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MMUL(X,Y,N1,N2,N3,Z)
      DIMENSION X(1),Y(1),Z(1)
      COMMON/MAIN1/NDIM
      N1M1=N1-1
      NN3=N3*NDIM
      DO 1 I=1,NN3,NDIM
      II=I+N1M1
      DO 1 J=I,II
1     Z(J)=0.0
      ENTRY MMULS
      NN3=N3*NDIM
      KK=0
      DO 10 K=1,N2
      DO 8 I=1,N1
      C1=X(I+KK)
      IF(C1.NE.0.0) CALL VADD1(NN3,C1,Z(I),Y(K))
8     CONTINUE
10    KK=KK+NDIM
      RETURN
      END
```

```
      SUBROUTINE DIAG2(N,A,B,C1,C2)
C     A = C1*B + C2*I
C     A,B ARE N*N MATRICES; I IS N*N IDENTITY MATRIX
      DIMENSION A(1), B(1)
      COMMON /MAIN1/ NDIM, NDIM1
      NN=N*NDIM
      NM1=N-1
      II=1
      IF (C1 .EQ. 1.0) GO TO 10
      DO 5 J=1,NN,NDIM
      K=J+NM1
      DO 4 I=J,K
4     A(I)=C1*B(I)
      A(II)=A(II)+C2
5     II=II+NDIM1
      RETURN
10    DO 7 J=1,NN,NDIM
      K=J+NM1
      DO 6 I=J,K
6     A(I)=B(I)
      A(II)=A(II)+C2
7     II=II+NDIM1
      RETURN
      END
```

```
      SUBROUTINE IDENT(N,A,C1)
      DIMENSION A(1)
      COMMON/MAIN1/NDIM,NDIM1
      NN=N*NDIM
      II=1
      DO 1 I=1,N
      DO 2 J=I,NN,NDIM
2     A(J)=0.0
      A(II)=C1
1     II=II+NDIM1
      RETURN
      END
```

```
      SUBROUTINE EQUATE(A,B,NR,NC)
C     A=B
C     MATRIX EQUATE
      DIMENSION A(1), B(1)
      CALL MSCALE(A,B,NR,NC,1.0)
      RETURN
      END
```

```
      SUBROUTINE MSCALE(A,B,NR,NC,C1)
C     A=C1*B
C     A AND B MAY BE EQUIVALENT
      DIMENSION A(1), B(1)
      COMMON /MAIN1/ NDIM
      NN=NC*NDIM
      IF (C1 .EQ. 1.0) GO TO 10
      IF (C1 .EQ. 0.0) GO TO 20
      IF (C1 .EQ. -1.) GO TO 30
      DO 5 I=1,NR
      DO 5 J=I,NN,NDIM
5     A(J)=C1*B(J)
      RETURN
10    DO 15 I=1,NR
      DO 15 J=I,NN,NDIM
15    A(J)=B(J)
      RETURN
20    DO 25 I=1,NR
      DO 25 J=I,NN,NDIM
25    A(J)=0.0
      RETURN
30    DO 35 I=1,NR
      DO 35 J=I,NN,NDIM
35    A(J)=-B(J)
      RETURN
      END
```

```
      FUNCTION DOT(NR,A,B)
      DOUBLE PRECISION DDT1, DBLE
      DIMENSION A(1),B(1)
      DDT1=0.0D0
      IF (NR .LE. 0) GO TO 2
      DO 1 I=1,NR
1     DDT1=DDT1+DBLE(A(I)*B(I))
2     DOT=DDT1
      RETURN
      END
```

```
      FUNCTION DOT2(NN,A,B)
      DOUBLE PRECISION DDT2, DBLE
      DIMENSION A(1),B(1)
      COMMON /MAIN1/ NDIM
      DDT2=0.0D0
      IF (NN .LE. 0) GO TO 2
      DO 1 I=1,NN,NDIM
1     DDT2=DDT2+DBLE(A(I)*B(I))
2     DOT2=DDT2
      RETURN
      END
```

```
      FUNCTION DOT3(N,A,B)
      DOUBLE PRECISION DDT3, DBLE
      DIMENSION A(1),B(1)
      COMMON /MAIN1/ NDIM
      DDT3=0.0D0
      IF (N .LE. 0) GO TO 2
      II=1
      DO 1 I=1,N
      DDT3=DDT3+DBLE(A(II)*B(I))
1     II=II+NDIM
2     DOT3=DDT3
      RETURN
      END
```

```
      SUBROUTINE VADD(N,C1,A,B)
      DIMENSION A(1),B(1)
      DO 1 I=1,N
1     A(I)=A(I)+C1*B(I)
      RETURN
      END
```

```
      SUBROUTINE VADD1(NN,C1,A,B)
      DIMENSION A(1),B(1)
      COMMON/MAIN1/NDIM
      DO 1 I=1,NN,NDIM
1     A(I)=A(I)+C1*B(I)
      RETURN
      END
```

```
      SUBROUTINE VSCALE(X,Y,N,C1)
      DIMENSION X(1),Y(1)
      L=0
      IF(C1.EQ.1.0) GO TO 5
      IF(C1.EQ.0.0) GO TO 8
      IF(C1.EQ.-1.) GO TO 13
1     L=L+1
      X(L)=C1*Y(L)
      IF(L.LT.N) GO TO 1
      RETURN
5     L=L+1
      X(L)=Y(L)
      IF(L.LT.N) GO TO 5
      RETURN
8     L=L+1
      X(L)=0.0
      IF(L.LT.N) GO TO 8
      RETURN
13    L=L+1
      X(L)=-Y(L)
      IF(L.LT.N) GO TO 13
      RETURN
      END
```

```
      SUBROUTINE VMAT1(A,X,N1,N2,Y)
C     Y=AX
      DIMENSION A(1),X(1),Y(1)
      COMMON/MAIN1/NDIM
      DO 1 I=1,N1
      Y(I)=0.0
      II=I
      DO 1 J=1,N2
      Y(I)=Y(I)+A(II)*X(J)
1     II=II+NDIM
      RETURN
      END
```

```
      SUBROUTINE VMAT2(Z,A,X,N1,N2,Y)
C     Y=Z+AX
      DIMENSION A(1),X(1),Z(1),Y(1)
      COMMON/MAIN1/NDIM
      DO 1 I=1,N1
      Y(I)=Z(I)
      II=I
      DO 1 J=1,N2
      Y(I)=Y(I)+A(II)*X(J)
1     II=II+NDIM
      RETURN
      END
```

```
      FUNCTION XGAIN(TH,XM,XS)
      DIMENSION A(5)
      DATA A/.2258368,-.2521287,1.259695,-1.287822,.9406461/
      IF (TH.GT.0.) GO TO 2
      XGAIN=1.0
      RETURN
2     Y=XM
      NS=2
      IF(XS.LT.1.0E-10)XS=1.0E-10
      IF(Y.EQ.0.) NS=1
      ANS=0.
      RMS=XS**2+XM**2
      DO 1 I=1,NS
      Z=.707*(TH+Y)/XS
      TEMP=EXP(-Z**2)
      X=1./(1.+.327591*ABS(Z))
      P=X*((((A(5)*X+A(4))*X+A(3))*X+A(2))*X+A(1))*1.128379
      ERF=1.-P*TEMP
      IF (Z.LT.0.) ERF=-ERF
      ANS=ANS+(RMS+TH*Y)*(1.-ERF)-XS*Y*TEMP*.7975
1     Y=-Y
      XGAIN=ANS/RMS/FLOAT(NS)
      IF(XGAIN.LT.1.E-6) XGAIN=1.E-6
      RETURN
      END
```

```
      SUBROUTINE MATIO(X,NR,NC,IO)
C     BATCH ORIENTED MATRIX I/O
C     IO=1   INPUT ONLY
C     IO=2   INPUT AND OUTPUT
C     IO=3   OUTPUT ONLY
C     IO=4   PUNCH

      DIMENSION X(1)
      COMMON /MAIN1/ NDIM
      COMMON /INOU/  KIN, KOUT, KPTR, KPUNCH

      JEND=NC*NDIM
      GO TO (5,5,20,40) IO

C*******INPUT
5     DO 10 I=1,NR
      READ (KIN,1000) (X(IJ), IJ=I,JEND,NDIM)
10    CONTINUE
      IF (IO .EQ. 1) RETURN

C*******OUTPUT
20    DO 30 I=1,NR
      WRITE (KOUT,2000) (X(IJ), IJ=I,JEND,NDIM)
30    CONTINUE
      RETURN

C*******PUNCH
40    DO 50 I=1,NR
      WRITE (KPUNCH,3000) (X(IJ),IJ=I,JEND,NDIM)
50    CONTINUE
      RETURN

1000  FORMAT (8E10.0)
2000  FORMAT (1H ,1P10E13.3)
3000  FORMAT (1P8E10.3)
      END
```

```
      SUBROUTINE VECTIO(X,N,IO)
C     BATCH ORIENTED VECTOR I/O
C     IO=1   INPUT ONLY
C     IO=2   INPUT AND OUTPUT
C     IO=3   OUTPUT ONLY
C     IO=4   PUNCH

      DIMENSION X(1)
      COMMON /INOU/  KIN, KOUT, KPTR, KPUNCH

      GO TO (10,10,20,40) IO

C*******INPUT
10    READ (KIN,1000) (X(I), I=1,N)
      IF (IO .EQ. 1) RETURN

C*******OUTPUT
20    WRITE (KOUT,2000) (X(I), I=1,N)
      RETURN

C*******PUNCH
40    WRITE (KPUNCH,3000) (X(I), I=1,N)
      RETURN

1000  FORMAT (8E10.0)
2000  FORMAT (1H ,1P10E13.3)
3000  FORMAT (1P8E10.3)
      END
```

```
      SUBROUTINE PAGEFD(KFIL,KOUNT)
C     WRITES KOUNT FORMFEEDS (1 IN COL 1) ON FILE KFIL

      ENTRY FORMFD
100   IF (KOUNT.LE.0) RETURN

      DO 200 I=1,KOUNT
      WRITE (KFIL,1000)
1000  FORMAT (1H1)
200   CONTINUE

      RETURN
      END
```

```
      SUBROUTINE DAYTIM(KFIL)
C     WRITES THE DATE AND THE TIME ON FILE KFIL

10    CALL TIME(LTIME)
      CALL DATE(LDATE)
      WRITE (KFIL,1000) LDATE, LTIME
1000  FORMAT(1H ,A10,2X,A10)
      RETURN

      END
```

DISTRIBUTION

	No. of Copies
Assistant Secretary of Defense for Manpower and Reserve Affairs Department of Defense Washington, D.C. 20301	1
HQ DA DAPE-PBR, Mr. Barber Washington, D.C. 20310	1
Commander US Army Materiel Development and Readiness Command Attn: DRXAM-TL 5001 Eisenhower Avenue Alexandria, Virginia 22333	2
Headquarters US Army Medical R&D Command Attn: Behavior Sciences Rsch Br. Main Navy Building Washington, D.C. 20315	1
Director Walter Reed Army Institute of Research Walter Reed Army Medical Center Attn: Neuropsychiatry Division Washington, D.C. 20012	1
OAD/E&LS ODDR&E, Pentagon, Room 3D129 Attn: LTC Henry L. Taylor Washington, D.C. 20310	1
US Army Human Engineering Laboratory Representative Attn: DRXHE-AV, Dr. M. Hofmann Directorate for RD&E US Army Aviation Systems Command P. O. Box 209 St. Louis, Missouri 63166	1
Commander US Naval Weapons Laboratory Attn: Technical Library Dahlgren, Virginia 22448	1
Commander US Naval Weapons Center Attn: Code 533, Technical Library China Lake, California 93555	1
Director, Development Center US Marine Corps Development and Education Command Attn: Chairman, Air Operations Division Quantico, California 22134	1
HQ Rome Air Development Center Attn: TILD Griffiss Air Force Base New York 13440	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-AM, Mr. Miles US Army Systems Analysis Activity Aberdeen Proving Ground Maryland 21005	1

	No of Copies
Chief Human Engineering Laboratory Detachment Attn: DRXHE-FA, Mr. Sorin Frankford Arsenal Philadelphia Pennsylvania 19137	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-PTA, Mr. Carlock Picatinny Arsenal Dover, New Jersey 07801	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-RI, Mr. Galbauy Rock Island Arsenal Rock Island, Illinois 61201	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-TA, Mr. J. Erickson US Army Tank-Automotive Research and Development Command Warren, Mississippi 48090	1
Chief Human Engineering Laboratory Detachment Attn: DRXHE-ME, Mr. Antenucci, c/o DRXFB-TO US Army Mobility Equipment Research and Development Command Fort Belvoir, Virginia 22060	1
US Army Research Office Attn: Information Processing Office Box CM, Duke Station Durham, North Carolina 27706	1
Commander US Army Air Defense School Attn: ATSA-CTD-MO Fort Bliss, Texas 79916	1
Commander Rock Island Arsenal Attn: SARRI-LP-L, Technical Library Rock Island, Illinois 61201	1
Commander US Army Electronics Command Attn: DRSEL-PA-RH, Mr. Baron Fort Monmouth, New Jersey 07703	1
Library US Army War College Carlisle Barracks Pennsylvania 17013	1
Director US Army Ballistic Research Laboratory Attn: DRXBR-EB Aberdeen Proving Ground Maryland 21005	1

	No. of Copies
Army Research Institute Field Unit Attn: Dr. Robert W. Bauer Building 2423 Fort Knox, Kentucky 40121	1
Department of Transportation Library Reference and Research Branch, TAD-494.6 800 Independence Avenue SW Washington, D.C. 20591	1
US Postal Service Laboratory Attn: Mr. D.Y. Cornog Chief, HF Group 11711 Parklawn Drive Rockville, Maryland 20852	1
Mr. Edgar M. Johnson US Army Research Institute Room 239, The Commonwealth Building 1320 Wilson Boulevard Arlington, Virginia 22209	1
Federal Aviation Administration Civil Aeromedical Institute Library AC-101.1, P. O. Box 25082 Oklahoma City, Oklahoma 73125	1
Department of Operations Analysis Naval Postgraduate School Attn: Professor James K. Arima Monterey, California 93940	1
Commander USAVSCOM Attn: DRSAV-R-F, Mr. S. Moreland P. O. Box 209 St. Louis, Missouri 63166	1
Director Naval Research Laboratory Attn: Code 5132A Washington, D.C. 20390	1
Commanding Officer Naval Training Equipment Center Attn: Technical Library Orlando, Florida 32813	1
6J70 AMRL Attn: MRHE MRHE, Dr. M. J. Warrick MRHER, Mr. C. Bates, Jr. Wright-Patterson Air Force Base Ohio 45433	 2 1 1
AF Flight Dynamics Laboratory Attn: FDCR (CDIC) AFFDC-FCR, LT Creg Peters Wright-Patterson Air Force Base Ohio 45433	 1 1

	No. of Copies
Director US Army Human Engineering Laboratory Attn: DRXHE-DBD Aberdeen Proving Cround Maryland 21005	2
Medical Library, Building 148 Naval Submarine Medical Research Laboratory Box 900, Submarine Base New London Croton, Connecticut 06340	1
Code 455 Office of Naval Research Washington, D.C. 20360	1
Dr. Marshall J. Farr Associate Director, Personnel and Training Code 458 Office of Naval Research Washington, D.C. 20360	1
Commandant US Army Artillery and Missile School Attn: USAAMS Technical Library Fort Sill, Oklahoma 73503	1
ARI Field Unit P. O. Box 6057 Fort Bliss, Texas 79916	1
Commander Fort Huachuca Support Command, US Army Attn: Tech References Division Fort Huachuca, Arizona 85613	1
Commander Attn: Technical Library White Sands Missile Range New Mexico 88002	1
Director USA Air Mobility Research and Development Laboratory Attn: Dr. Richard S. Dunn Ames Research Center Moffett Field, California 94035	1
Commander US Army Electronics Command Attn: DRSEL-VL-E Fort Monmouth, New Jersey 07703	1
Director Military Psychology and Leadership United States Military Academy West Point, New York 10996	1
Commander Watervliet Arsenal Attn: SWEWV-RDT Watervliet, New York 12189	1
Commander Frankford Arsenal Attn: Library (C2500, Building 51-2) Philadelphia, Pennsylvania 19137	1

	No. of Copies
ARI Field Unit Attn: Library Fort Knox, Kentucky 40121	1
Commander US Army Tank-Automotive Command Attn: DRSTA-RHFL, Research Library Warren, Mississippi 48090	1
Commander US Army Tank-Automotive Command Attn: DRSTA-R Warren, Mississippi 48090	1
Director of Graduate Studies and Research Attn: Behavioral Sciences Representative US Army Command and General Staff College Fort Leavenworth, Kansas 66027	1
US Army Agency for Aviation Safety Attn: Librarian Fort Rucker, Alabama 36360	1
Chief Army Research Institute P. O. Box 476 Fort Rucker, Alabama 36360	1
Commander US Army Aeromedical Research Laboratory P. O. Box 577 Attn: Dr. Kent Kimball	1
Library	1
Fort Rucker, Alabama 36360	
US Army Natick Research and Development Command Attn: Tech Library (DRXNM-STL) Natick, Massachusetts 01760	1
US Army Natick Research and Development Command Behavioral Sciences Division Attn: DRXNM-PRB	1
DRXNM-PRBE	1
Natick, Massachusetts 01760	
US Army Medical Bioengineering Research and Development Laboratory Fort Detrick, Building 568 Frederick, Maryland 21701	1
Commander US Army Training Device Agency, Naval Training Equipment Center Attn: Code N2A Orlando, Florida 32813	1

	No. of Copies
US Army Infantry School Library Infantry Hall Fort Benning, Georgia 31905	1
Project Manager Office of the Project Manager for Training Devices Attn: DRCPM-TND, Dr. R.E. Odom Fort Benning, Georgia 31905	1
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12
Commander US Army Materiel and Readiness Command Attn: DRCRD	1
DRCDL	1
5001 Eisenhower Avenue Alexandria, Virginia 22333	
Bolt, Beranek and Newman Inc. Attn: Dr. S. Baron	2
Dr. D. Kleinmas	1
Dr. J. Baliser	1
50 Moulton Street Cambridge, Maryland 02138	
DRXHE-MI, Mr. Chaikin	1
DRCPM-HFE	1
-AH	1
-MPE	1
-LDE	1
-ROL-E	1
-RK	1
-HEL-T	1
-TOE	1
-MW	1
DRSMI-LP, Mr. Voigt	1
-W	1
DRDMI-X, Dr. McDaniel	1
-T, Dr. Kobler	1
-H	1
-Y	1
-YD	1
-YT	1
-TD	1
-TDC	1
-TDD	1
-TDF	1
-TDK	1
-TDR	1
-TDS	1
-TDW	1
-TDW, Mr. Dickson	5
-TE	1
-TG	1
-TL	1
-TPN, L. Aymett	1
-TBD	3
-TI (Record Set)	1
(Reference Copy)	1